

Titre: Séquencement de primitives pour la synthèse de mouvements
Title: naturels en robotique

Auteur: Alexandre Willame
Author:

Date: 2014

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Willame, A. (2014). Séquencement de primitives pour la synthèse de
Citation: mouvements naturels en robotique [Mémoire de maîtrise, École Polytechnique de
Montréal]. PolyPublie. <https://publications.polymtl.ca/1640/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/1640/>
PolyPublie URL:

**Directeurs de
recherche:** Jérôme Le Ny, David Saussié, & Richard Gourdeau
Advisors:

Programme: génie électrique
Program:

UNIVERSITÉ DE MONTRÉAL

SÉQUENCEMENT DE PRIMITIVES POUR LA SYNTHÈSE DE MOUVEMENTS
NATURELS EN ROBOTIQUE

ALEXANDRE WILLAME
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)
DÉCEMBRE 2014

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

SÉQUENCEMENT DE PRIMITIVES POUR LA SYNTHÈSE DE MOUVEMENTS
NATURELS EN ROBOTIQUE

présenté par : WILLAME Alexandre

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. BARON Luc, Ph.D., président

M. LE NY Jérôme, Ph.D., membre et directeur de recherche

M. SAUSSIÉ David, Ph.D., membre et codirecteur de recherche

M. GOURDEAU Richard, Ph.D., membre et codirecteur de recherche

M. ACHICHE Sofiane, Ph.D., membre

DÉDICACE

”Le véritable voyage de découverte ne
consiste pas à chercher de nouveaux
paysages, mais à avoir de nouveaux
yeux.”

Marcel Proust

REMERCIEMENTS

Je tiens à remercier tous ceux et celle qui ont contribué de près ou de loin à ce mémoire. Je remercie tout particulièrement mes directeurs de recherche, Jérôme Le Ny, David Saussié et Richard Gourdeau, de qui les remarques et conseils continuels ont permis de m'améliorer et de faire de ce sujet de recherche ce qu'il est.

Je remercie les gens de la Moment Factory, en particulier Quentin Bleton et Julie choinière qui m'ont permis de travailler avec eux et dont l'expérience restera encore longtemps gravée dans ma mémoire.

Je remercie Hervé Jeanmart et Paul Fisette de l'École Polytechnique de Louvain qui m'ont permis de réaliser cet échange avec l'École Polytechnique de Montréal et qui m'aura appris bien des choses, sur le plan universitaire, professionnel et personnel.

Je tient également à remercier le personnel du département de génie électrique. Marc Charboneau, Marie-Lyne Brisson, Nathalie Lévesque, Suzanne Le Bel, Ghislaine Ethier-Carrier, pour leur aide.

Je remercie le Conseil de Recherches en Sciences Naturelles et en génie du Canada (CRSNG-NSERC) pour l'octroi du financement qui a permis de mener à bien cette recherche.

Je remercie enfin les étudiants du département avec qui j'ai pu partager de bons moments et qui m'ont permis d'évoluer dans un environnement amical. Vincent Dubanchet, Caroline Dietrich, Antoine Berthelemot, Erik Hailer, Jacques Mbous Ikong, Jacques Michiels, Benoît Le Berr, Meisam Mohammadi et tous ceux que j'aurais pu oublier.

Ma dernière pensée va à ma famille, mes amis et mes proches qui m'ont accompagné durant cette aventure. Au-delà de ce document, je leur dédie le fruit de ce travail.

Alexandre Willame
Montréal, le 26 janvier 2015

RÉSUMÉ

Générer des mouvements complexes d'apparence naturelle est désirable dans de nombreuses applications homme-robot, mais reste une tâche difficile à réaliser en temps réel. Dans ce mémoire, nous proposons un système génératif pour le séquençement interactif de mouvements naturels pour un bras manipulateur. Basé sur l'enchaînement de mouvements élémentaires, le système a été développé afin de séquencer des trajectoires d'apparence naturelle tout en minimisant les temps de calcul, permettant la reproduction de tels mouvements dans un environnement avec contraintes de temps.

Nous séquençons des mouvements élémentaires, exprimés comme des trajectoires à minimum de jerk, dérivée troisième de la position, afin de réexprimer les multiples problèmes d'optimisation originaux en un nouveau problème unique et simplifié. Le système permet de générer une nouvelle trajectoire plus complexe dont l'apparence reste naturelle tout le long. Le problème d'optimisation complet est résolu en un temps relativement court, et peut être résolu en temps réel selon le cas. La naturalité des mouvements obtenus est étudiée et comparée à d'autres techniques.

Les primitives de mouvement utilisées ne sont pas reproduites exactement dans la solution finale ; elles sont influencées par les mouvements antérieurs et postérieurs, ce qui correspond à l'observation qu'un être humain ne répète jamais deux fois le même mouvement parfaitement. L'implémentation d'un contrôle pour ces trajectoires est décrite pour deux robots manipulateurs industriels différents à six degrés de liberté. Le contrôle a dû être adapté par rapport au mode de contrôle idéal pour une trajectoire naturelle afin de satisfaire aux modes de contrôle disponibles pour ces robots.

Trois applications sont présentées : la première permet de reproduire l'écriture humaine avec un bras manipulateur ; la seconde permet d'enchaîner des mouvements enregistrés au préalable à l'aide d'un dispositif de capture de mouvements. Enfin, une preuve de concept montre un robot manipulateur réalisant une chorégraphie synchronisée sur le rythme d'une musique.

ABSTRACT

Generating complex natural-looking movements is desirable in a variety of human-robot interaction scenarios, but remains a task that is challenging to perform in real-time. In this presentation, we describe a generative system for sequencing natural movements for robotic manipulators. Based on the sequencing of movement primitives, the system is developed to generate natural-looking trajectories while minimizing computational time. This allows for the reproduction of natural movements in an environment with time constraints.

We sequence movement primitives, expressed using the minimum jerk model, so as to express the various optimization problems into a unique and simpler optimization problem. The system thus allows for the generation of a new, more complex trajectory with a natural looking appearance. The optimization problem is solved in a relatively short amount of time and can be solved in real time in some cases. The natural appearance of the resulting trajectories is studied and compared to other techniques.

The final motion does not exactly reproduce the primitives that it is made of; it depends on the movements that occur before and after it. This mimics the natural behaviour of a human that never reproduces the exact same movement twice. The control has been adapted to two different industrial robots so that it can suit the available control mode of such robots.

Three applications are presented: the first reproducing human writing with a robotic manipulator. The second allowing to sequence motion recorded using a motion capture device. The last application consists of a choreography of a robot waving a flag in synchronisation with the beat of a music.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
LISTE DES ANNEXES	xvi
LISTE DES SIGLES ET ABRÉVIATIONS	xvii
CHAPITRE 1 INTRODUCTION	1
1.1 Mise en contexte	1
1.2 Objectifs de recherche	2
1.3 Plan du mémoire	3
CHAPITRE 2 REVUE DE LITTÉRATURE	6
2.1 Définition d'un mouvement naturel	6
2.2 Planification de mouvements naturels	8
2.2.1 Méthodes d'optimisation	8
2.2.2 Paramétrisation du profil de vitesse	9
2.2.3 Autres types de planification de trajectoires naturelles	9
2.2.4 Choix du critère de naturalité	9

2.3	Reproduction de trajectoire planifiée	10
2.4	Utilisation de primitives de mouvements	10
CHAPITRE 3 Modèle du Minimum Jerk		12
3.1	Trajectoire point à point	13
3.2	Trajectoire passant par n points, t_i fixés (MJTF)	17
3.3	Trajectoire passant par n points, t_i optimisés (MJTO)	24
3.4	Comparaison entre MJTF et MJTO	26
3.5	Comparaison entre MJ et DL	27
3.6	Comparaison avec le blending	31
3.6.1	Blending de deux trajectoires avec $\alpha(t)$, $\beta(t)$ linéaires	32
3.6.2	Blending de deux trajectoires avec $\alpha(t)$, $\beta(t)$ constants	34
3.7	Conclusion	37
CHAPITRE 4 Composition Séquentielle		39
4.1	Principe de la composition séquentielle	39
4.2	Optimisation de la trajectoire complète	41
4.3	Transition entre primitives de mouvements	42
4.4	Résultats	44
4.5	Timescaling	50
4.6	Conclusion	50
CHAPITRE 5 Prise en considération du domaine articulaire		52
5.1	Cinématique inverse analytique	52
5.1.1	FANUC LR Mate 200iC	53
5.1.2	ABB IRB 120T	53
5.1.3	Choix de la solution analytique	53
5.2	Cinématique inverse itérative : Resolved Motion Rate	60
5.3	Alternance des types de cinématique inverse	60
5.4	Mélange de MJ cartésien et articulaire	63

5.5	Conclusion	68
CHAPITRE 6	Mise en oeuvre sur un contrôleur industriel	69
6.0.1	Robot FANUC	70
6.0.2	Robot ABB	74
CHAPITRE 7	Applications	75
7.1	Écriture de lettres manuscrites	75
7.2	Reproduction de captures de mouvements	78
7.3	Chorégraphie sur une musique	78
7.3.1	Implémentation générale	80
7.3.2	Interface avec X-Agora	80
7.3.3	Considérations techniques	83
7.3.4	Preuve de concept	87
7.3.5	Discussion	89
CHAPITRE 8	CONCLUSION	91
8.1	Synthèse des travaux	91
8.2	Limitations de la solution proposée et améliorations futures	91
8.3	Conclusion	93
ANNEXES	100
A.1	Découplage du problème de positionnement et orientation	101
A.2	Cinématique inverse du FANUC LR Mate 200iC	103
A.3	Cinématique inverse du ABB IRB 120T	110
B.1	Vitesses et accélérations	113
D.1	RMR - Évitement des limites articulaires	119
D.2	RMR - Évitement des singularités	119
E.1	Équations de la dynamique	121
E.2	Dynamique inverse : calcul des forces articulaires	122

E.3	Dynamique directe : calcul des accélérations articulaires	122
E.4	Détermination des paramètres dynamiques	125
F.1	Notions de base	127
F.2	Interpolation de quaternions	127
F.3	Passage des angles d'Euler aux quaternions	130
F.4	Passage des quaternions aux angles d'Euler	133

LISTE DES TABLEAUX

Tableau 1.1	Introduction : récapitulatif des différentes techniques présentées dans ce mémoire	5
Tableau 5.1	Cinématique inverse : cinématique inverse du FANUC LR Mate 200iC .	54
Tableau 5.2	Cinématique inverse : cinématique inverse du ABB IRB 120T	54
Tableau 5.3	Cinématique inverse : limites articulaires	56
Tableau 5.4	Cinématique inverse : erreurs pour l'exemple de la figure 5.4	58
Tableau 7.1	Application : tableau des commandes LUA et RAPID associées	86
Tableau A.1	Cinématique inverse : paramètres de DH pour un poignet sphérique . .	102
Tableau A.2	Cinématique inverse : paramètres de DH du FANUC LR Mate 200iC .	103
Tableau A.3	Cinématique inverse : cinématique inverse du FANUC LR Mate 200iC .	109
Tableau A.4	Cinématique inverse : paramètres de DH du ABB IRB 120T	110
Tableau A.5	Cinématique inverse : cinématique inverse du ABB IRB 120T	112
Tableau E.1	Dynamique : paramètres dynamiques du FANUC LR Mate 200iC . . .	125

LISTE DES FIGURES

Figure 2.1	Revue de littérature : principe de reproduction d'une trajectoire naturelle	7
Figure 2.2	Revue de littérature : comparaison entre le profil de vitesse d'une trajectoire naturelle et le profil de vitesse trapézoïdal	7
Figure 3.1	Naturalité : trajectoire point à point et ses conditions limites	13
Figure 3.2	Naturalité : trajectoire en 1 dimension obtenue par MJ	15
Figure 3.3	Naturalité : comparaison du MJ et du DL	16
Figure 3.4	Naturalité : illustration du problème de MJTF pour $n = 3$ points de passage avec temps intermédiaires fixés	19
Figure 3.5	Naturalité : exemple de paysage de coût du problème à $n = 3$ points et t_i fixé	23
Figure 3.6	Naturalité : illustration du problème de MJTO pour $n = 3$ points de passage	24
Figure 3.7	Naturalité : programmation séquentielle convexe	25
Figure 3.8	Naturalité : optimisation de t_1 pour $n = 3$ points de passage	27
Figure 3.9	Naturalité : comparaison de trajectoire avec et sans optimisation de t_i .	28
Figure 3.10	Naturalité : comparaison des optimisations de MJTF et MJTO	29
Figure 3.11	Naturalité : comparaison entre MJ et DL pour une trajectoire point à point	30
Figure 3.12	Naturalité : comparaison entre MJ et DL pour l'écriture de 'hms' . . .	30
Figure 3.13	Naturalité : principe de base du blending	31
Figure 3.14	Naturalité : utilisation du blending	32
Figure 3.15	Naturalité : comparaison entre le blending et le modèle de MJTF . . .	36
Figure 3.16	Naturalité : comparaison entre le blending et le modèle de MJTO . . .	37
Figure 4.1	Composition séquentielle : principe	39
Figure 4.2	Composition séquentielle : technique	40
Figure 4.3	Composition séquentielle : Illustration de la récupération de points de passage	42

Figure 4.4	Composition séquentielle : fonction de coût J_V	43
Figure 4.5	Composition séquentielle : forme de la fonction de coût des transitions .	44
Figure 4.6	Composition séquentielle : exemple typique de la fonction de coût pour $\alpha = 0.95$	44
Figure 4.7	Composition séquentielle : mouvement résultant pour la fonction de coût de la figure 4.6	45
Figure 4.8	Composition séquentielle : exemple	46
Figure 4.9	Composition séquentielle : Jerks pour l'exemple de la figure 4.8	47
Figure 4.10	Composition séquentielle : temps de calcul pour l'exemple de la figure 4.8	47
Figure 4.11	Composition séquentielle : autre exemple	48
Figure 4.12	Composition séquentielle : Jerks pour l'exemple de la figure 4.8	49
Figure 4.13	Composition séquentielle : temps de calcul pour l'exemple de la figure 4.11	49
Figure 5.1	Cinématique inverse : repères du FANUC LR Mate 200iC	53
Figure 5.2	Cinématique inverse : ABB IRB 120T	55
Figure 5.3	Cinématique inverse : $q(t_{i-1})$ est pris sans limites articulaires	57
Figure 5.4	Cinématique inverse : trajectoire pour l'algorithme hors-ligne de sélection de trajectoires	58
Figure 5.5	Cinématique inverse : choix de la condition initiale	59
Figure 5.6	Cinématique inverse : trajectoire pour la cinématique inverse mixte . .	61
Figure 5.7	Cinématique inverse : ω_{ps} pour une trajectoire d'exemple avec CI mixte	62
Figure 5.8	Cinématique inverse : alternance de CI analytique et itérative	62
Figure 5.9	Cinématique inverse : comparaison des temps de calcul entre la CI analytique, itérative et mixte	62
Figure 5.10	Cinématique inverse : optimisation mixte du MJ cartésien et MJ articulaire	63
Figure 5.11	Cinématique inverse : exemple de trajectoire à MJ hybride	67
Figure 6.1	Contrôle : principe du contrôle point à point avec un robot industriel .	70
Figure 6.2	Contrôle : profil de vitesse avec CNTxx	71
Figure 6.3	Contrôle : profil de vitesse avec CNTxx	71

Figure 6.4	Contrôle : reproduction d'une trajectoire avec paramètre $CNT = 80$. .	72
Figure 6.5	Contrôle : reproduction d'une trajectoire avec paramètre $CNT = 100$.	73
Figure 6.6	Contrôle : principe du zonedata	74
Figure 6.7	Contrôle : profil de vitesse avec Zonedata	74
Figure 7.1	Application : librairies de primitives de lettres	75
Figure 7.2	Application : reproduction d'écriture manuscrite	76
Figure 7.3	Application : reproduction d'écriture manuscrite	77
Figure 7.4	Application : librairies de primitives de MoCaps converties	78
Figure 7.5	Application : reproduction de mouvements de MoCap	79
Figure 7.6	Application : reproduction de mouvements de MoCap	79
Figure 7.7	Application : librairie de mouvements pour la preuve de concept	80
Figure 7.8	Application : schéma de fonctionnement du système élaboré pour la preuve de concept	81
Figure 7.9	Application : suite de logiciels X-Agora	82
Figure 7.10	Application : exemple de fichier XML contenant une trajectoire	83
Figure 7.11	Application : exemple de découpage d'une commande par paquets pour le protocole TCP	83
Figure 7.12	Application : envoi de commande en Lua au serveur X-Agora	84
Figure 7.13	Application : réception de paquets TCP en RAPID	84
Figure 7.14	Application : commande de déplacement du robot	85
Figure 7.15	Application : modification de la résolution temporelle d'un mouvement	87
Figure 7.16	Application : timeline X-Agora pour la preuve de concept	88
Figure 7.17	Application : preuve de Concept	88
Figure 7.18	Application : exemple d'extension du robot synchronisé pour une ap- plication avec projection mapping	90
Figure A.1	Cinématique inverse : découplage avec un poignet sphérique	102
Figure A.2	Cinématique inverse : référentiel de l'effecteur	103
Figure A.3	Cinématique inverse : repères du FANUC LR Mate 200iC	104
Figure A.4	Cinématique inverse : repères du FANUC LR Mate 200iC	110

Figure C.1	Cinématique inverse : choix de ω_{cond} pour le DLS	118
Figure E.1	Dynamique inverse : termes pour NER	122
Figure E.2	Dynamique : schéma Simmechanics	126
Figure F.1	Quaternions : principe du SLERP	128

LISTE DES ANNEXES

Annexe A	Cinématique inverse	101
Annexe B	Vitesses et accélérations articulaires	113
Annexe C	Damped Least-Squares	115
Annexe D	RMR - Critères d'optimisations	119
Annexe E	Dynamique	121
Annexe F	Quaternions	127

LISTE DES SIGLES ET ABRÉVIATIONS

CAD	Computer-Aided Design
CI	Cinématique inverse
CTC	Computed torque control (contrôle à couple pré-calculé)
DH	Denavit-Hartenberg
DDL	Degrés De Liberté
DL	Delta-Lognormal
DLS	Damped Least Square (moindres carrés amortis)
MoCap	Motion Capture
MJ ou MJTO	Minimum Jerk (avec Temps intermédiaires Optimisés)
MJTF	Minimum Jerk avec temps intermédiaires fixés
NER	Newton Euler Recursif (Newton Euler Recursive)
RMR	Resolved Motion Rate
SCP	Sequential Convex Programming (programmation séquentielle convexe)
SL	Sigma-Lognormal

CHAPITRE 1

INTRODUCTION

1.1 Mise en contexte

Séquencer des mouvements complexes d'apparence naturelle est désirable dans de nombreuses applications homme-robot, mais reste une tâche difficile à réaliser en temps réel.

Un système peut nécessiter de générer une trajectoire sans connaître longtemps à l'avance des informations sur celle-ci. Par exemple, un robot fonctionnant au rythme d'une musique inconnue devra s'adapter au fur et à mesure des changements de tempo de la musique, sans avoir d'information sur ces changements à l'avance.

La reproduction de trajectoires naturelles est un problème rencontré dans de nombreux domaines. On retrouve cela dans le domaine médical (par exemple le récent robot de réhabilitation Jaco de Kinova Robotics (Maheu et al., 2011)), et dans l'animation. Dans le monde industriel, il a été prouvé qu'utiliser des trajectoires souples pour des robots manipulateurs permet de diminuer sa consommation énergétique (Hirakawa and Kawamura, 1996).

Dans le monde du spectacle, deux techniciens français ont créé depuis 2008 des chorégraphies avec des robots industriels (Audouin and Stegemann, 2013). Le but est de faire danser de tels robots sur scène. Les robots utilisés sont des robots de marque Staubli et chaque mouvement est paramétré à la main en utilisant le langage VAL3 propre au robot. Selon les créateurs, il faut compter un jour de travail pour programmer 10 secondes de chorégraphie pour un robot.

Dans le domaine du cinéma, des bras spécifiquement conçus pour les tournages (camera dollys) sont utilisés pour filmer des scènes. Certains d'entre eux sont automatisés et permettent une synchronisation avec des éléments extérieurs. Bot & Dolly est une compagnie à l'origine de différentes vidéos tournées avec un robot industriel, et ont notamment réalisé le tournage du film Gravity sorti en 2013. Il s'agit d'une des premières tentatives sérieuses d'utilisation de robots industriels dans le monde du cinéma (Pescovitz, 2014) (Seymour, 2013). La technique utilisée consiste à générer hors ligne la trajectoire à reproduire par le robot et à la rejouer lors du tournage. Les trajectoires en elles-mêmes n'ont pas été pensées afin de paraître naturel ;

l'apport d'un tel travail consiste à adapter un robot industriel à un tel environnement et à sa synchronisation avec d'autres éléments extérieurs.

Le travail présenté ici vise à réaliser un système permettant d'assembler et de synchroniser des mouvements naturels élémentaires entre eux pour un bras manipulateur en gardant l'apparence naturelle de ces mouvements. Le but est de pouvoir réaliser des chorégraphies synchronisées avec des éléments extérieurs (musique, projecteurs, autres dispositifs).

Le projet de recherche présenté dans ce mémoire a été effectué dans le cadre d'un accord de double diplôme entre l'École Polytechnique de Louvain (Belgique) et l'École Polytechnique de Montréal (Canada). Il s'est déroulé entièrement à l'École Polytechnique de Montréal. Un stage de deux mois a été effectué dans la compagnie montréalaise Moment Factory afin de réaliser une preuve de concept.

OBJECTIF

1.2 Objectifs de recherche

L'objectif de cette recherche est de concevoir un système permettant la génération de trajectoires naturelles pour un bras manipulateur dans le cadre d'applications interactives pour plusieurs cas de scénario. Le système devra fournir des outils permettant la génération de trajectoires naturelles utilisables aussi bien dans le cas d'applications presque temps réel, c-a-d. hors-ligne mais dans un temps court, ainsi que dans le cas d'applications interactives en temps réel. Par temps court, on entend un temps acceptable pour un concepteur de chorégraphie multimédia afin de pouvoir concevoir rapidement des esquisses convenables de chorégraphies, par opposition aux techniques hors ligne plus coûteuses en temps de calcul. On estime ce temps à moins d'une dizaine de secondes de calcul pour une petite chorégraphie constituée de 5 primitives de mouvement ou moins. Une utilisation typique de tels outils couvre la création de chorégraphies pour un bras manipulateur et la réaction interactive d'un robot à des facteurs imprévisibles longtemps à l'avance.

La plupart des algorithmes de génération de trajectoires naturelles sont complexes et demandent un certain temps de calcul, ce qui ne permet pas d'applications interactives. Un des objectifs de cette recherche est de fournir un outil permettant de s'affranchir de cette contrainte. Nous supposons qu'un mouvement naturel minimise le critère du minimum jerk (MJ), qui sera ainsi prit comme critère de naturalité pour une trajectoire donnée. Une trajectoire sera jugée plus naturelle qu'une autre si elle minimise mieux le critère du MJ. Ce

critère est expliqué au chapitre 2.

L’approche utilisée est d’assembler des mouvements naturels élémentaires entre eux, en permettant certaines modifications élémentaires sur ces mouvements de base, afin de générer des mouvements plus complexes. L’apparence de ces mouvements doit rester naturelle, c-a-d minimisant au mieux le critère de MJ. On veut également que les mouvements élémentaires possèdent une certaine composante aléatoire, c-a-d. que chaque mouvement élémentaire ne soit pas reproduit exactement de la même façon si répété.

Les transitions entre primitives sont calculées de telle sorte que l’on peut former de nouvelles trajectoires plus complexes et qui gardent l’apparence naturelle sous sa forme globale.

1.3 Plan du mémoire

Nous développons tout d’abord une revue de littérature à propos de la reproduction de trajectoires naturelles et de l’approche par primitives de mouvements en robotique en chapitre 2. Nous détaillons ensuite au chapitre 3 le critère de naturalité choisi pour décrire un mouvement naturel, et comparons les résultats obtenus à d’autres techniques du domaine, en terme de naturalité et en temps de calcul.

Nous décrivons au chapitre 4 le système de la composition séquentielle développé. Nous y traitons son principe, l’aspect naturel des trajectoires générées et discutons des temps de génération. Nous comparons les caractéristiques de ce système aux autres modèles de génération de trajectoires existantes en terme de temps de calcul et de naturalité.

Nous rappelons au chapitre 5 la cinématique inverse analytique et itérative. Nous analysons les différences entre la génération de trajectoires dans le domaine cartésien et dans le domaine articulaire du robot, et l’impact sur la naturalité du mouvement obtenu. Nous présentons également une technique de génération de trajectoires naturelles mélangeant optimisation dans les domaines cartésien et articulaire.

Les contrôleurs industriels étant relativement fermés quand il en vient à modifier le comportement bas-niveau des robots, nous décrivons au chapitre 6 les modes de contrôle adoptés et discutons des limitations de tels contrôleurs pour la reproduction de mouvements naturels.

Nous présentons enfin au chapitre 7 différents résultats pratiques obtenus en laboratoire avec une imitation d’écriture humaine par un robot manipulateur, la reproduction de trajectoires

obtenues par capture de mouvements (Motion Capture), ainsi qu'une application réalisée aux studios de Moment Factory pour le cas de chorégraphies synchronisées sur une musique dans un environnement logiciel adéquat.

Le tableau 1.1 décrit les différentes techniques décrites dans ce mémoire, un exemple d'application pour chacune, et le chapitre qui leur est associé.

Tableau 1.1 Tableau récapitulatif des différentes techniques présentées dans ce mémoire et exemple de scénario dans le cas de création d'une chorégraphie.

Exemple de scénario	Contraintes de temps	Besoin en terme de qualité du mouvement	Technique	Chapitre
Mouvement point à point entre primitives de mouvement	Forte (temps réel)	Intermédiaire	MJTF	3.2
Planification long terme de réalisation d'une chorégraphie	Aucune	Optimale (cartésien ou articulaire)	MJTO	3.3
Aperçu rapide d'une chorégraphie lors de sa conception	Faible (entre temps réel et cas du MJTO)	Intermédiaire (Assemblage de mouvements optimisés)	Composition séquentielle avec temps optimisés	4
Applications interactives	Forte (temps réel)	Intermédiaire (Assemblage de mouvements optimisés)	Composition séquentielle avec temps fixés	4
Objet fixé au bout du robot à déplacer dans le domaine cartésien et optimisation simultanée de l'apparence des mouvements du bras du robot	Aucune	optimisée (cartésien et articulaire)	Optimisation mixte dans le domaine cartésien et articulaire	5

CHAPITRE 2

REVUE DE LITTÉRATURE

Ce chapitre définit ce qu'est un mouvement naturel, dresse l'état de l'art sur les théories permettant de reproduire de tels mouvements et décrit les techniques utilisées en robotique permettant d'atteindre ce but.

La section 2.1 définit la notion de mouvement naturel pour un mouvement de bras chez l'homme. La section 2.2 décrit les différentes techniques de planification de trajectoires naturelles. Le choix de la méthode de génération de trajectoires naturelles est discuté en section 2.2.4. Les différents modes de contrôle pour la reproduction d'une trajectoire naturelle sont décrits en section 2.3.

2.1 Définition d'un mouvement naturel

Shadmehr and Mussa-Ivaldi (1994) et Flash (1987) ont découvert que la reproduction d'un mouvement naturel se déroule en deux étapes : d'abord une phase de planification de la trajectoire à reproduire, puis une phase de reproduction effective de la trajectoire planifiée (Fig. 2.1). Si le contrôleur utilisé pour la reproduction est comparable à la façon dont le bras humain est contrôlé, le problème de naturalité se situe au niveau de la planification de trajectoires.

Mais avant tout, il s'agit de comprendre ce que l'on entend par *naturalité* d'un mouvement. Dans notre cas, nous nous restreindrons à la naturalité de mouvements pour un bras humain. Un même chemin peut être perçu comme naturel lorsqu'il est parcouru à différentes vitesses. Un mouvement point à point pourra par exemple paraître naturel à basse ou haute vitesse, selon le cas. On oppose souvent un mouvement humain naturel (fluide, lisse) à un mouvement abrupt, typique des robots industriels. Ceux-ci utilisent un profil de vitesse trapézoïdal pour des trajectoires point à point, avec une accélération maximale, un palier de vitesse constant suivi d'une décélération maximale. Le profil de vitesse entre deux points pour une trajectoire naturelle est un profil en cloche légèrement asymétrique. Les deux profils sont illustrés en figure 2.2 (Plamondon and Srihari, 2000).

Les différentes techniques de planification de mouvements naturels décrites dans les prochaines sections approximent toutes des mouvements naturels ; le critère strict permettant de

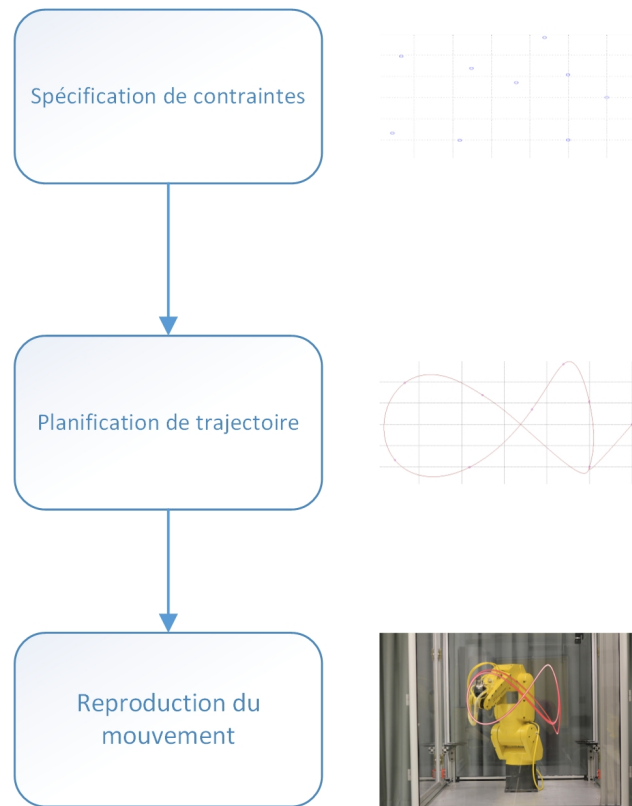


Figure 2.1 Principes de reproduction d'une trajectoire naturelle (adapté de Viviani and Flash (1995)).

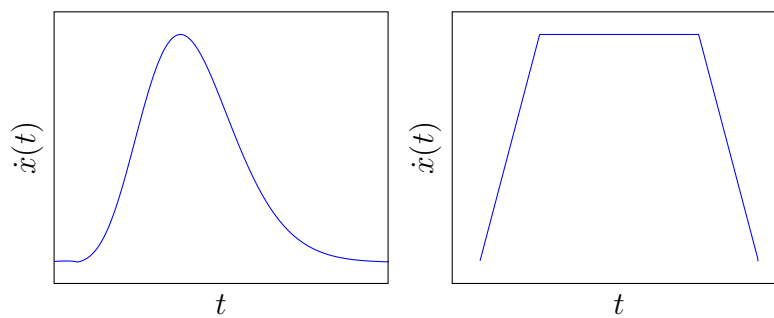


Figure 2.2 Comparaison entre le profil de vitesse d'une trajectoire naturelle et le profil de vitesse trapézoïdal. Le profil de vitesse naturel a une forme en cloche légèrement asymétrique pour des trajectoires point à point.

quantifier la naturalité d'une trajectoire dépend de la technique choisie. Chaque technique définit de par sa nature ce qu'est un mouvement naturel. On en conclura qu'avant tout, un mouvement est naturel s'il est perçu comme naturel auprès d'autres personnes ; il s'agit d'un test analogue au test de Turing afin de juger du caractère humain de la trajectoire réalisée.

2.2 Planification de mouvements naturels

2.2.1 Méthodes d'optimisation

L'étude du mouvement d'un bras humain au niveau fondamental (de l'étude musculaire et neuronale à l'étude du mouvement en tant que trajectoire) touche au domaine de la neuroscience. M. Kawato résume bien la base du problème et l'approche par optimisation que l'on retrouve dans un grand nombre de techniques pour la génération de trajectoires (Kawato, 1996). Il explique notamment l'origine de l'optimisation de certains critères comme le minimum de *jerk* (dérivée de l'accélération) :

« The minimum jerk criterion does not produce acceleration jumps at the start and end points, while the minimum acceleration criterion does. Based on this observation, some researchers suppose that the human brain implicitly adopts smoothness as the criterion for motion planning. »

Cette interprétation d'un mouvement naturel comme étant un mouvement lisse se retrouve implicitement dans toutes les techniques de génération de trajectoires naturelles : le modèle Delta-Lognormal (DL) de Plamondon and Djoua (2006) superpose par exemple plusieurs profils de vitesse qui ont la forme d'exponentielles paramétrées.

De nombreux critères d'optimisations autres que le MJ, existent : minimum d'accélération, jerk snap, torque, tension de muscles (Kawato, 1996). Selon les indications de Flash and Hogan (1985), Shadmehr and Mussa-Ivaldi (1994), le minimum de jerk apparaît comme le critère d'optimisation le plus adapté à la reproduction de trajectoires naturelles. Il est de plus également utilisé en robotique afin de minimiser les vibrations pour des robots industriels.

Parallèlement à cela, Shadmehr and Mussa-Ivaldi (1994) et Flash (1987) ont mis en évidence que la planification de la tâche s'effectue dans le plan cinématique, indépendamment des conditions dynamiques du système. Ces derniers ont également trouvé que la planification pour une trajectoire point à point s'effectue dans le domaine articulaire du bras.

2.2.2 Paramétrisation du profil de vitesse

Dans le domaine de la reconnaissance d'écriture, (Plamondon and Djoua, 2006) ont décrit les modèles de profils de vitesse Delta Lognormal (DL) et, plus récemment, Sigma-Lognormal (SL), qui consistent en un ensemble de paramètres permettant de définir l'action des muscles agonistes et antagonistes d'un bras humain afin de reproduire le plus fidèlement possible le profil de vitesse de mouvements rapides. Cette technique est particulièrement efficace dans le domaine de la reconnaissance d'écriture et la caractérisation de l'état du système nerveux de l'individu reproduisant une trajectoire.

La technique est plus adaptée à la reproduction de trajectoires qu'à la génération de trajectoires à proprement parler : l'extraction des paramètres du profil de vitesse se fait à partir de trajectoires de référence et en utilisant des algorithmes génétiques (Djoua et al., 2007). Ceux-ci ne fonctionnent pas en temps réel et prennent un temps de calcul conséquent.

Le profil de vitesse observé chez l'être humain pour une trajectoire point à point est une cloche asymétrique (Plamondon and Srihari, 2000) (Wada and Kawato, 1995).

2.2.3 Autres types de planification de trajectoires naturelles

À partir d'un apprentissage sur un grand nombre de données, Kim and Park (2008) et Kwon and Park (2008) ont développé une structure logicielle permettant de générer des trajectoires à partir d'une base de données de mouvements en utilisant un modèle de Markov caché. Kulic et al. (2008), Kulic et al. (2011), et Khatib et al. (2009) utilisent une approche similaire afin de classifier des primitives de mouvements, et permettre leur identification en temps réel lors de leur reproduction par un être humain. La technique semble correctement élaborée mais demande d'enregistrer un grand nombre de mouvements d'un être humain, ce qui rends la génération de nouvelles primitives de mouvement fastidieuse et nécessite un matériel d'enregistrement adapté.

2.2.4 Choix du critère de naturalité

Toutes les techniques présentées dans les sections précédentes dépendent d'une façon ou d'une autre de paramètres imposés arbitrairement permettant d'approcher au mieux les caractéristiques estimées d'une trajectoire naturelle.

Dans le cas du MJ, la durée du mouvement est un paramètre. Si on demande un grand mouvement pour une durée faible (plus que ce que peut faire un être humain), on obtiendra systématiquement un mouvement non naturel, bien que le critère d'optimisation soit res-

pecté pour de telles contraintes (Pattacini et al., 2010). La question de définir ce qu'est un mouvement naturel en terme de critère d'optimisation revient donc à trouver un critère où le concept de naturalité est complètement inclus dans le critère d'optimisation et ne dépend pas des paramètres d'entrée.

Les mêmes remarques s'appliquent aux techniques de paramétrisation du profil de vitesse ainsi qu'aux autres techniques décrites ci-dessus. De plus, ces dernières dépendent de trajectoires naturelles de référence afin de pouvoir approximer au mieux cette trajectoire. Dans ce sens, elles sont plus adaptées à la reconnaissance ou la reproduction de trajectoires naturelles existantes qu'à leur génération.

Nous avons besoin d'une technique permettant de générer en temps réel de nouvelles trajectoires. Nous choisissons d'utiliser le modèle du MJ comme critère de naturalité, qui nous permettra de générer de nouveaux mouvements, et dont la formulation comme un problème d'optimisation et se prête particulièrement bien à une adaptation en vue d'une exécution temps réel comme nous allons le voir au prochaine chapitre. Il s'agit de plus d'un modèle utilisé dans certains domaines de la robotique afin de minimiser l'énergie consommée que les vibrations (Yoshioka et al., 2011).

2.3 Reproduction de trajectoire planifiée

Flash (1987) a prouvé que l'être humain exécute un mouvement en deux étapes, avec d'abord une planification dans le domaine articulaire une reproduction avec un système de contrôle assimilable à un système de masse-amortisseur.

Ijspeert et al. (2002) et Ijspeert (2008) ont élaboré un système de contrôle non linéaire dynamique permettant de contrôler un bras robotisé agissant de façon similaire tout en étant robuste face à des perturbations extérieures. Si le bras rencontre un obstacle, le système retarde la consigne afin que le bras évite de forcer en cas de collision. Avec ce système, une trajectoire est générée selon le MJ afin d'obtenir des coefficients pour le système dynamique pour reproduire la trajectoire (Shadmehr and Wise, 2005).

2.4 Utilisation de primitives de mouvements

L'idée de séquencer des primitives de mouvement ensemble afin de générer des mouvements plus complexes en robotique n'est pas nouvelle ; il s'agit d'un concept de plus en plus exploité afin de pallier la complexité des mouvements à reproduire.

En robotique, Schaal et al. (2005) utilisent des primitives de mouvements dynamiques qui

permettent de caractériser le système dynamique non linéaire permettant de contrôler le bras d'un robot humanoïde. Woch and Plamondon (2004) ont discuté de cela dans le cas précis de l'écriture avec le modèle DL. On retrouve le concept de primitives de mouvements dans de nombreux autres travaux, notamment le *Motion Description Language* (MDL) (Brockett, 1990), (Hristu-Varsakelis et al., 2003), le *pre-image backchaining* (Burridge et al., 1999), et le *maneuver automaton* de Frazzoli et al. (2005).

Notre approche de la composition de mouvements est quelque peu différente de ces techniques dans le sens où ces techniques se basent sur la caractérisation de coefficients de contrôleur et le passage d'un contrôleur à un autre (Le Ny and Pappas, 2012). Notre approche décomposant la planification du contrôle, la composition de mouvement se déroule directement au niveau de la planification de la trajectoire. Ainsi, elle se rapproche plus de ce qui se fait dans le domaine de l'animation par ordinateur, en particulier les *Motion Graphs* de Kovar et al. (2002) et les *Parametric Motion Graphs* de Heck and Gleicher (2007). Il s'agit d'une approche par graphe de mouvements paramétrés, où une trajectoire complexe peut être formée à partir de plusieurs mouvements de base paramétrés et assemblés les uns avec les autres. Ces mouvements sont assemblés en interpolant la fin et le début des deux mouvements adjacents.

Enfin, Hersch and Billard (2008), Pattacini et al. (2010) et Flacco et al. (2012) ont réalisé un contrôleur pour robot manipulateur permettant de réaliser des trajectoires point à point respectant un critère hybride dans le domaine articulaire aussi bien que cartésien. De façon analogue, en se basant sur les observations de Shadmehr and Mussa-Ivaldi (1994) et Flash (1987) décrites plus haut, nous développons en section 5.4 un système de MJ hybride afin de générer une trajectoire naturelle respectant simultanément un critère dans le domaine articulaire et cartésien.

CHAPITRE 3

Modèle du Minimum Jerk

Une trajectoire $x(t) \in \mathbb{R}^3$ remplissant le critère du MJ (Flash and Hogan, 1985) minimise le critère

$$J(x(t)) = \frac{1}{2} \int_{t_0}^{t_f} \left\| \left(\frac{d^3 x(t)}{dt^3} \right) \right\|^2 dt, \quad (3.1)$$

où t_0 est le temps initial, t_f le temps final, et $\|\cdot\|$ la norme euclidienne. Sans perte de généralité, on prendra par la suite $t_0 = 0$.

La trajectoire $x(t)$ est de dimension 3. Flash and Hogan (1985) démontrent que le jerk s'optimise indépendamment dans chaque axe pour le problème de positionnement. Le critère de l'Eq. 3.1 peut ainsi s'écrire

$$\begin{aligned} J(x(t)) &= \frac{1}{2} \int_0^{t_f} \left(\frac{d^3 x_x(t)}{dt^3} \right)^2 + \left(\frac{d^3 x_y(t)}{dt^3} \right)^2 + \left(\frac{d^3 x_z(t)}{dt^3} \right)^2 dt \\ &= \frac{1}{2} \int_0^{t_f} \left(\frac{d^3 x_x(t)}{dt^3} \right)^2 dt + \frac{1}{2} \int_0^{t_f} \left(\frac{d^3 x_y(t)}{dt^3} \right)^2 dt + \frac{1}{2} \int_0^{t_f} \left(\frac{d^3 x_z(t)}{dt^3} \right)^2 dt \\ &= J(x_x(t)) + J(x_y(t)) + J(x_z(t)). \end{aligned}$$

L'analyse du MJ se fera dans ce document dans une seule dimension par simplicité de notation ; le problème réel consiste en l'optimisation indépendante de 3 problèmes similaires avec des contraintes d'égalité différentes.

Le problème est contraint par n points de passages par lesquels la trajectoire doit passer. Nous étudions tout d'abord en section 3.1 le cas classique dans la littérature d'une trajectoire point à point. Nous regarderons ensuite la généralisation du problème pour une trajectoire passant par n points, d'abord pour un problème simplifié avec temps intermédiaires t_i fixés en section 3.2, puis pour le problème complet de MJ en section 3.3. Nous étudierons également la naturalité, en terme du critère du MJ, des trajectoires générées.

3.1 Trajectoire point à point

Étudions tout d'abord le cas d'une trajectoire point à point avec durée du mouvement t_f imposé. Les contraintes générales pour une trajectoire point à point sont :

$$\begin{cases} x(0) = x_0; & x(t_f) = x_f \\ \dot{x}(0) = \dot{x}_0; & \dot{x}(t_f) = \dot{x}_f \\ \ddot{x}(0) = \ddot{x}_0; & \ddot{x}(t_f) = \ddot{x}_f, \end{cases} \quad (3.2)$$

avec $x_0 \in \mathbb{R}^3$ la position initiale et $x_f \in \mathbb{R}^3$ la position finale. La situation est illustrée en figure 3.1.

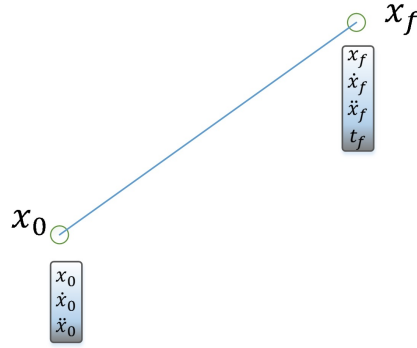


Figure 3.1 Trajectoire point à point et ses conditions limites.

En utilisant le calcul des variations, on peut démontrer qu'une condition nécessaire pour que la trajectoire qui respecte (3.1) et (3.2) est (Shadmehr and Wise, 2005) (Kirk, 2004)

$$x^{(6)} = 0. \quad (3.3)$$

L'ensemble des fonctions satisfaisant le critère (3.3) est donc l'ensemble des polynômes d'ordre 5 fonction du temps t de la forme

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5, \quad (3.4)$$

et vérifiant les conditions d'égalité 3.2, ce qui permet d'exprimer les conditions sur le polynôme

$x(t)$

$$\begin{cases} x(0) = x_0 = a_0 \\ \dot{x}(0) = \dot{x}_0 = a_1 \\ \ddot{x}(0) = \ddot{x}_0 = 2a_2 \\ x(t_f) = x_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\ \dot{x}(t_f) = \dot{x}_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\ \ddot{x}(t_f) = \ddot{x}_f = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3. \end{cases}$$

Les coefficients valent alors

$$\begin{cases} a_0 = x_0 \\ a_1 = \dot{x}_0 \\ a_2 = \frac{\ddot{x}_0}{2} \\ a_3 = -\frac{3\ddot{x}_0 t_f^2 - \ddot{x}_f t_f^2 + 12t_f \dot{x}_0 + 8t_f \dot{x}_f + 20x_0 - 20x_f}{2t_f^3} \\ a_4 = -\frac{-3\ddot{x}_0 t_f^2 + 2\ddot{x}_f t_f^2 - 16t_f \dot{x}_0 - 14t_f \dot{x}_f - 30x_0 + 30x_f}{2t_f^4} \\ a_5 = -\frac{\ddot{x}_0 t_f^2 - \ddot{x}_f t_f^2 + 5t_f \dot{x}_0 + 6t_f \dot{x}_f + 12x_0 - 12x_f}{2t_f^5}. \end{cases} \quad (3.5)$$

Dans ce cas-ci, les coefficients sont complètement déterminés par les contraintes d'égalité. Si on reprend les contraintes sous forme matricielle, on a

$$Ea = d \quad (3.6)$$

avec

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{pmatrix}; a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix}; d = \begin{pmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \\ x_f \\ \dot{x}_f \\ \ddot{x}_f \end{pmatrix}.$$

La solution existera toujours si la matrice carrée E est non-singulière, c-a-d. si son déterminant est non nul. Ici, la matrice E est définie positive. En effet, son déterminant vaut

$$\det(E) = 4t_f^9$$

et ses valeurs propres sont strictement positives $\forall t_f > 0$.

Pour le cas particulier où on part et on arrive au repos ($\dot{x}_0 = \dot{x}_f = \ddot{x}_0 = \ddot{x}_f = 0$), qui est le cas bien connu et souvent traité dans la littérature pour comparer les modèles de mouvements naturels (Viviani and Flash, 1995) (Plamondon and Srihari, 2000) (Flash and Hogan, 1985), on a

$$\begin{cases} a_0 = x_0 \\ a_1 = 0 \\ a_2 = 0 \\ a_3 = -\frac{20x_0 - 20x_f}{2t_f^3} \\ a_4 = -\frac{-30x_0 + 30x_f}{2t_f^4} \\ a_5 = -\frac{12x_0 - 12x_f}{2t_f^5}, \end{cases} \quad (3.7)$$

Une trajectoire typique est illustrée en figure 3.2.

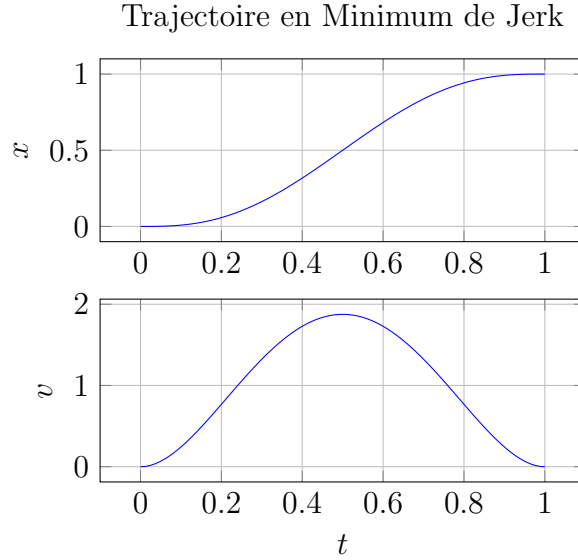


Figure 3.2 Trajectoire 1D point à point obtenue par MJ. $x_0 = 0$, $x_f = 1$ et $t_f = 1$.

On compare en figure 3.3 une trajectoire point à point obtenue avec le MJ et le DL. On observe que les deux profils de vitesse sont proches, au détail près que le profil de vitesse DL est seul capable de reproduire l'asymétrie d'un profil de vitesse, comme démontré par Plamondon and Srihari (2000). Il s'agit d'une observation importante à faire : le MJ n'est pas une technique permettant de reproduire au mieux une trajectoire naturelle, il existe d'autres techniques plus précises. Nous utiliserons néanmoins ici le MJ étant donné sa forme qui nous permettra plus loin de simplifier grandement le problème et d'améliorer les temps de calculs.

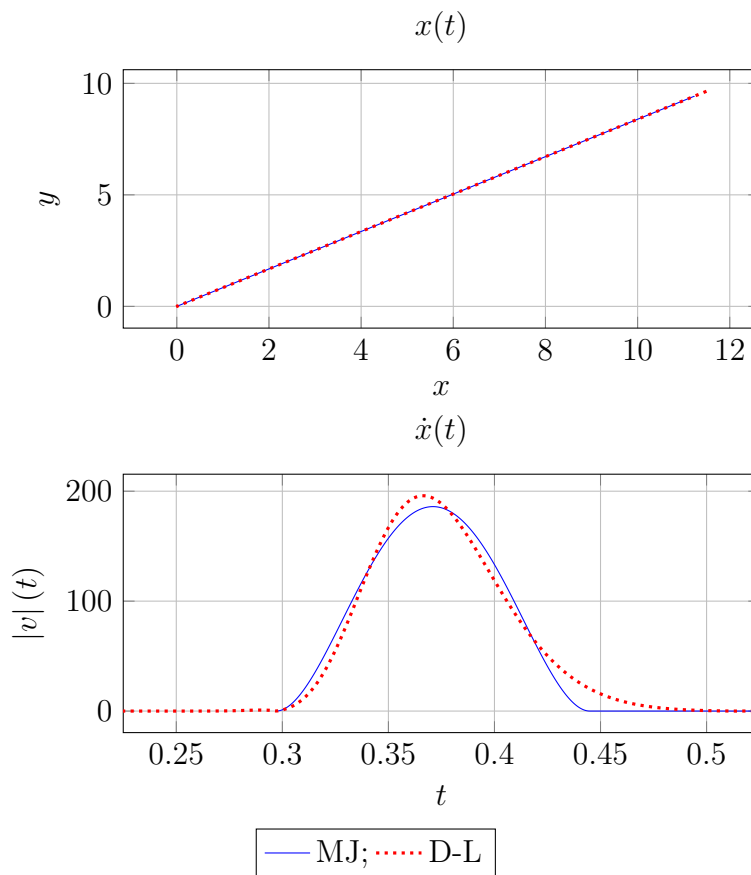


Figure 3.3 Comparaison du MJ et du DL pour une trajectoire point à point. Le profil de vitesse DL arrive à reproduire l'asymétrie typique d'un profil de vitesse naturel. Celui généré par le MJ est proche du DL malgré sa symétrie.

3.2 Trajectoire passant par n points, t_i fixés (MJTF)

Une trajectoire longue est en général plus complexe qu'un mouvement entre deux points. Après le cas point à point étudié à la section précédente, penchons-nous sur le cas général à n points. La durée totale t_f du mouvement est imposée. Nous supposons pour le moment que les temps de passage intermédiaire t_i sont imposés. Il s'agit d'un problème de Minimum Jerk avec temps intermédiaires fixés (MJTF) ; il ne s'agit pas du problème de Minimum Jerk complet, qui sera vu plus loin.

On fixe les temps de passage comme étant une pondération de t_f selon les distances euclidiennes entre chaque point de passage (Fig. 3.4).

$$t_i = \frac{|x_i - x_{i-1}|}{\sum_{j=1}^{n-1} |x_j - x_{j-1}|} t_f, \quad (3.8)$$

Une trajectoire de MJ passant par n points est représentée par la concaténation de $n - 1$ polynômes consécutifs (Flash and Hogan, 1985), (Piazzi and Visioli, 2000)

$$x(t) = \begin{cases} x_0(t) & \text{si } 0 \leq t < t_1 \\ x_1(t) & \text{si } t_1 \leq t < t_2 \\ \vdots & \\ x_{n-1}(t) & \text{si } t_{n-1} \leq t < t_f \end{cases}$$

Nous allons ici, de façon semblable à ce qui a été fait en section 3.1, substituer le temps initial au temps final pour chaque polynôme $x_i(t)$ afin d'alléger l'écriture. Nous écrivons pour chaque polynôme $x_i(t)$,

$$\Delta t_i = t_i - t_{i-1},$$

en imposant $t_i = 0 \forall \Delta t_i$ de telle sorte que

$$\begin{aligned} x_i(0) &= x_i \\ x_i(\Delta t_{i+1}) &= x_{i+1}. \end{aligned}$$

Exprimons le critère du MJ 3.1 en terme des coefficients des polynômes

$$J(x_i(t)) = \frac{1}{2} \left(36\Delta t_i a_{i3}^2 + 144\Delta t_i^2 a_{i3} a_{i4} + 192\Delta t_i^3 a_{i4}^2 + 240\Delta t_i^3 a_{i3} a_{i5} + \right. \\ \left. 720\Delta t_i^4 a_{i4} a_{i5} + 720\Delta t_i^5 a_{i5}^2 \right),$$

avec a_{ij} le $j^{\text{ème}}$ coefficient du polynôme de $x_i(t)$. Il s'agit de trouver les polynômes formant la nouvelle trajectoire $x(t)$ respectant le critère du MJ

$$J(x(t)) = \sum_{i=0}^{n-2} \frac{1}{2} \int_0^{\Delta t_{i+1}} \left(\frac{d^3 x_i(t)}{dt^3} \right)^2 dt. \quad (3.9)$$

Par rapport aux contraintes d'égalité du problème point à point de l'Eq. 3.2, nous avons, en plus des positions, vitesses et accélérations des points extrêmes, la continuité des positions, vitesses et accélérations aux points intermédiaires, ainsi que les positions intermédiaires, c-a-d.

$$\left\{ \begin{array}{ll} x_0(0) & = x_0; \\ \dot{x}_0(0) & = \dot{x}_0; \\ \ddot{x}_0(0) & = \ddot{x}_0; \\ x_0(\Delta t_1) & = x_1; \\ \dot{x}_0(\Delta t_1) & = \dot{x}_1(0) \\ \ddot{x}_0(\Delta t_1) & = \ddot{x}_1(0) \\ x_1(0) & = x_1; \\ x_1(\Delta t_2) & = x_2; \\ \dot{x}_1(\Delta t_2) & = \dot{x}_2(0) \\ \ddot{x}_1(\Delta t_2) & = \ddot{x}_2(0) \\ \vdots & \vdots \\ x_{n-2}(\Delta t_{n-1}) = x_{n-1}(0) & = x_{n-2}; \\ \dot{x}_{n-2}(\Delta t_{n-1}) & = \dot{x}_{n-1}(0) \\ \ddot{x}_{n-2}(\Delta t_{n-1}) & = \ddot{x}_{n-1}(0) \\ x_{n-1}(\Delta t_f) & = x_f; \\ \dot{x}_{n-1}(\Delta t_f) & = \dot{x}_f; \\ \ddot{x}_{n-1}(\Delta t_f) & = \ddot{x}_f; \end{array} \right. , \quad (3.10)$$

La situation est illustrée en figure 3.4 pour $n = 3$ points. À noter que la vitesse et l'accélération aux points intermédiaires ne sont pas connues, seule leur continuité est imposée. La continuité jusqu'aux accélérations est nécessaire pour avoir un minimum à l'Eq. 3.9 car sinon, le jerk serait infini à la jonction des polynômes.

La fonction de coût (3.9) s'écrit de façon générale sous la forme matricielle

$$J = \frac{1}{2} \mathbf{a}^\top \mathbf{Q} \mathbf{a} \quad (3.11)$$

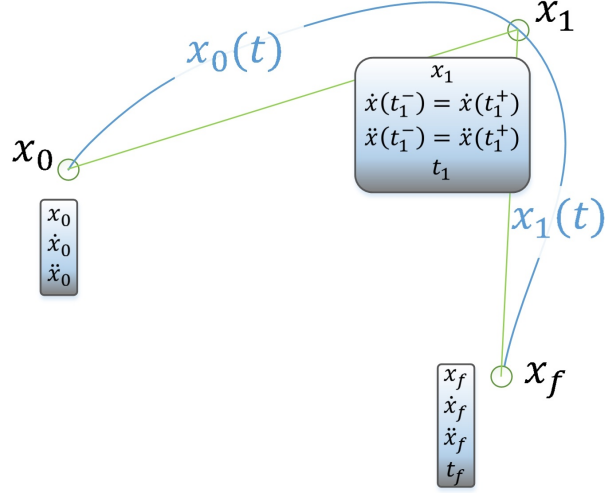


Figure 3.4 Illustration du problème de MJTF pour $n = 3$ points de passage avec t_i fixés. Les traits en vert sont les distances euclidiennes entre les points de passage permettant de calculer la valeur initiale des temps de passage t_i .

avec les contraintes (3.10) exprimées de façon matricielle

$$\mathbf{E}\mathbf{a} = \mathbf{d}. \quad (3.12)$$

Les Eqs. 3.11 et (3.12) forment un problème quadratique linéaire.

Sa résolution peut se faire en utilisant les multiplicateurs de Lagrange. De façon générale, on a la fonction de coût augmentée

$$H(\mathbf{a}) = J(\mathbf{a}) + \lambda^\top \mathbf{f}(\mathbf{a})$$

avec $\mathbf{f}(\mathbf{a}) = \mathbf{E}\mathbf{a} - \mathbf{d} = 0$ au minimum, et λ est le vecteur de multiplicateurs de Lagrange. Les conditions nécessaires pour avoir un point stationnaire, et possiblement minimum, sont

$$\begin{aligned} \frac{\partial H}{\partial \mathbf{a}} &= 0 \\ \frac{\partial H}{\partial \lambda} &= \mathbf{f}(\mathbf{a}) = 0. \end{aligned}$$

Pour que le point trouvé soit un minimum, une condition suffisante est que (Kirk, 2004)

$$\frac{\partial^2 H}{\partial \mathbf{a}^2} = Q > 0, \quad (3.13)$$

qui signifie que la fonction est concave au point minimum. Dans notre cas, nous avons la fonction de coût augmentée à minimiser

$$\begin{aligned} H(\mathbf{a}) &= \frac{1}{2} \mathbf{a}^\top Q \mathbf{a} + \lambda^\top (E \mathbf{a} - \mathbf{d}) \\ \frac{\partial H}{\partial \mathbf{a}} &= \mathbf{0} = Q \mathbf{a} + E^\top \lambda \\ \frac{\partial H}{\partial \lambda} &= \mathbf{0} = E \mathbf{a} - \mathbf{d}. \end{aligned}$$

On obtient le système linéaire augmenté

$$\begin{pmatrix} Q & E^\top \\ E & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{d} \end{pmatrix} \quad (3.14)$$

qu'on appelle système KKT (Karush–Kuhn–Tucker) pour le problème d'optimisation quadratique avec contraintes linéaires (Boyd and Vandenberghe, 2004, chap. 10). Il est constitué de

$$\begin{aligned} Q &= \begin{pmatrix} Q_1 & 0 & \dots & 0 & 0 \\ 0 & Q_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & Q_{(n-2)} & 0 \\ 0 & 0 & \dots & 0 & Q_{(n-1)} \end{pmatrix} \\ Q_i &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 36\Delta t_i & 72\Delta t_i^2 & 120\Delta t_i^3 \\ 0 & 0 & 0 & 72\Delta t_i^2 & 192\Delta t_i^3 & 360\Delta t_i^4 \\ 0 & 0 & 0 & 120\Delta t_i^3 & 360\Delta t_i^4 & 720\Delta t_i^5 \end{pmatrix} = \begin{pmatrix} 36t_f & 72\Delta t_i^2 & 120\Delta t_i^3 \\ 72\Delta t_i^2 & 192\Delta t_i^3 & 360\Delta t_i^4 \\ 120\Delta t_i^3 & 360\Delta t_i^4 & 720\Delta t_i^5 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned} \quad (3.15)$$

Q étant une matrice symétrique, Q_i la matrice propre au polynôme i et semi-définie positive,

et \otimes est le produit de Kronecker. On exprime également les contraintes d'égalité (3.12)

$$E\mathbf{a} = \mathbf{d}$$

$$E = \begin{pmatrix} E_1 & 0 & 0 & \dots & 0 & 0 \\ E_{2a} & E_{2b} & 0 & \dots & 0 & 0 \\ 0 & E_{3a} & E_{3b} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & E_{(n-1)a} & E_{(n-1)b} \\ 0 & 0 & 0 & \dots & 0 & E_n \end{pmatrix}$$

avec

$$\begin{aligned} E_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & \Delta t_i & \Delta t_i^2 & \Delta t_i^3 & \Delta t_i^4 & \Delta t_i^5 \end{pmatrix} \\ (E_{ia}E_{ib}) &= \left(\begin{array}{cccccc|cccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t_i & \Delta t_i^2 & \Delta t_i^3 & \Delta t_i^4 & \Delta t_i^5 \\ 0 & 1 & 2\Delta t_{i-1} & 3\Delta t_{i-1}^2 & 4\Delta t_{i-1}^3 & 5\Delta t_{i-1}^4 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6\Delta t_{i-1} & 12\Delta t_{i-1}^2 & 20\Delta t_{i-1}^3 & 0 & 0 & -2 & 0 & 0 & 0 \end{array} \right) \\ E_n &= \begin{pmatrix} 0 & 1 & 2\Delta t_n & 3\Delta t_n^2 & 4\Delta t_n^3 & 5\Delta t_n^4 \\ 0 & 0 & 2 & 6\Delta t_n & 12\Delta t_n^2 & 20\Delta t_n^3 \end{pmatrix} \end{aligned}$$

avec $i \neq \{0, n\}$ pour (E_{ia}, E_{ib}) , et

$$\mathbf{d} = \begin{pmatrix} x_0 & \dot{x}_0 & \ddot{x}_0 & x_1 & \dot{x}_1 & x_2 & 0 & 0 & \dots & x_{n-2} & \dot{x}_{n-1} & 0 & 0 & \dot{x}_f & \ddot{x}_f \end{pmatrix}^\top.$$

avec $x_{n-1} = x_f$. On veut savoir si le point trouvé est un minimum global du problème. Une condition suffisante pour que la solution du système linéaire augmenté de l'Eq. 3.14 soit un minimum global est que la matrice KKT du système augmenté

$$\begin{pmatrix} Q & E^\top \\ E & \mathbf{0} \end{pmatrix} \quad (3.16)$$

soit toujours inversible (Boyd and Vandenberghe, 2004), c-a-d si son déterminant est non nul.

Prenons un exemple pour $n = 3$ points de passage. La matrice de l'Eq. 3.14 vaut

$$\begin{pmatrix} Q & E^\top \\ E & \mathbf{0} \end{pmatrix} = \left(\begin{array}{cccccccccccc|cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \Delta t_1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & \Delta t_1^2 & 0 & 0 & 2\Delta t_1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 36\Delta t_1 & 72\Delta t_1^2 & 120\Delta t_1^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t_1^3 & 0 & 0 & 3\Delta t_1^2 & 6\Delta t_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 72\Delta t_1^2 & 192\Delta t_1^3 & 360\Delta t_1^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t_1^4 & 0 & 0 & 4\Delta t_1^3 & 12\Delta t_1^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 120\Delta t_1^3 & 360\Delta t_1^4 & 72\Delta t_1^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t_1^5 & 0 & 0 & 5\Delta t_1^4 & 20\Delta t_1^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t_2 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t_2^2 & 0 & -2 & 2\Delta t_2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 36\Delta t_2 & 72\Delta t_2^2 & 120\Delta t_2^3 & 0 & 0 & 0 & 0 & \Delta t_2^3 & 0 & 0 & 3\Delta t_2^2 & 6\Delta t_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 72\Delta t_2^2 & 192\Delta t_2^3 & 360\Delta t_2^4 & 0 & 0 & 0 & 0 & \Delta t_2^4 & 0 & 0 & 4\Delta t_2^3 & 12\Delta t_2^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 120\Delta t_2^3 & 360\Delta t_2^4 & 72\Delta t_2^5 & 0 & 0 & 0 & 0 & \Delta t_2^5 & 0 & 0 & 5\Delta t_2^4 & 20\Delta t_2^3 & 0 & 0 \\ \hline 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & \Delta t_1 & \Delta t_1^2 & \Delta t_1^3 & \Delta t_1^4 & \Delta t_1^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t_2 & \Delta t_2^2 & \Delta t_2^3 & \Delta t_2^4 & \Delta t_2^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2\Delta t_1 & 3\Delta t_1^2 & 4\Delta t_1^3 & 5\Delta t_1^4 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6\Delta t_1 & 12\Delta t_1^2 & 20\Delta t_1^3 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2\Delta t_2 & 3\Delta t_2^2 & 4\Delta t_2^3 & 5\Delta t_2^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6\Delta t_2 & 12\Delta t_2^2 & 20\Delta t_2^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

Son déterminant vaut

$$\det \left[\begin{pmatrix} Q & E^\top \\ E & \mathbf{0} \end{pmatrix} \right] = 110592\Delta t_1^{14}\Delta t_2^{14}(\Delta t_1 + \Delta t_2)^4$$

et est non nul $\forall \Delta t_1, \Delta t_2 > 0$. De façon générale, on observe que toute matrice KKT (3.16) avec $n > 2$ et $\Delta t_i \neq 0$ (qui est une contrainte implicite du problème) a son déterminant non nul. Le problème possède donc toujours une solution unique.

On peut aussi voir que la matrice KKT est de rang plein (de par son déterminant non-nul), alors que la matrice Q seule est semi-définie positive, et donc pas de rang plein. Ceci signifie que l'ajout des contraintes d'égalité au problème d'optimisation initial (sans contrainte) contraint ce dernier de telle sorte que la solution du problème complet est unique. La solution obtenue est donc le minimum global pour le problème avec contraintes d'égalité.

On peut également voir ce problème comme un problème d'optimisation des vitesses et accélérations aux points de passage. En effet, il est intéressant d'observer que le problème possède $6(n-1)$ degrés de liberté (DDL). Les différentes contraintes imposées sont

- les contraintes de positions : n ;

- les contraintes de continuité de position : $(n - 2)$;
- les contraintes de continuité de vitesses : $(n - 2)$;
- les contraintes de continuité d'accélération : $(n - 2)$;
- les contraintes de vitesses aux limites : 2 ;
- et les contraintes d'accélérations aux limites 2,

pour un total de $2(n - 1)$ contraintes. Ceci nous laisse avec

$$6(n - 1) - 2(n - 1) = 2(n - 2)$$

DDL afin d'optimiser le jerk le long de la trajectoire. Ces DDL correspondent aux vitesses et aux accélérations aux points intermédiaires $x(t_i)$. On impose en effet la continuité de ces vitesses et accélérations mais pas leur valeur. La solution du système augmenté de l'Eq. 3.14 résout le système d'équations afin d'obtenir les vitesses et les accélérations intermédiaires optimales pour le problème de MJTF. Un exemple de paysage du coût du MJTF en fonction des vitesses et des accélérations intermédiaires $\dot{x}(t_1)$ et $\ddot{x}(t_1)$ est illustré en figure 3.5. On observe que le point optimal est bien un minimum du problème convexe pour ces vitesses et accélérations.

Nous utiliserons en pratique la fonction MATLAB `quadprog` pour résoudre le problème d'optimisation.

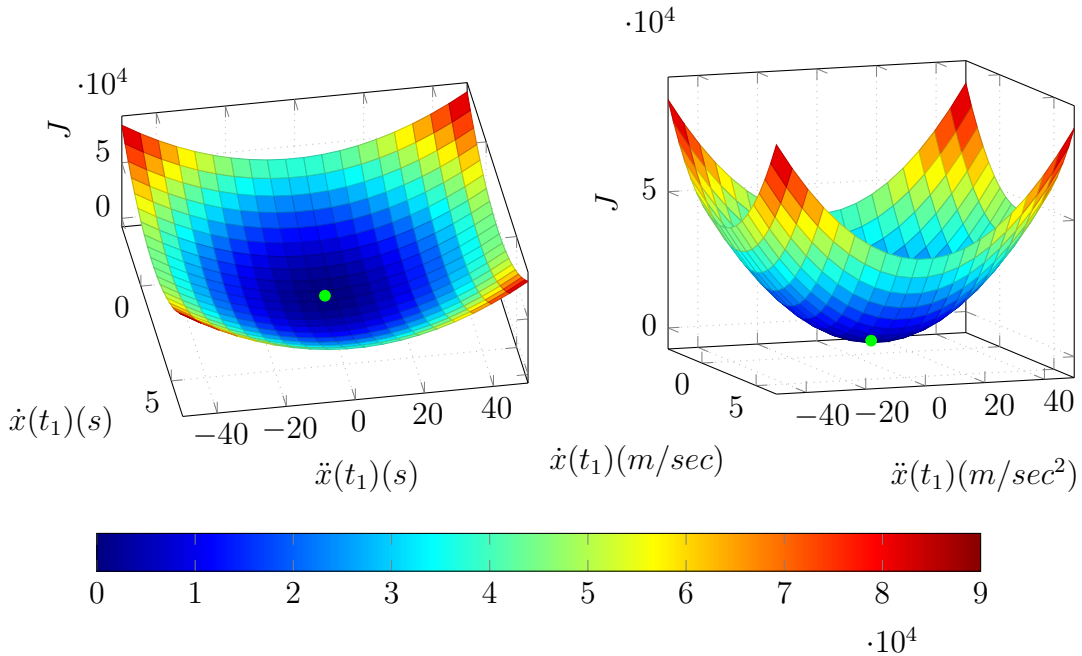


Figure 3.5 Exemple de paysage de coût du problème à $n = 3$ points et t_i fixé.

3.3 Trajectoire passant par n points, t_i optimisés (MJTO)

Nous avons supposé les temps de passage t_i constants jusqu'à maintenant. En spécifiant les points de passage x_i , il est en fait difficile de savoir le moment exact auquel la trajectoire passe par ces points. Nous allons maintenant optimiser les temps de passage t_i afin de minimiser le jerk. On suppose la durée totale t_f du mouvement constante. Il s'agit d'un problème de Minimum Jerk avec temps intermédiaires optimisés (MJTO), c-a-d. le vrai problème de Minimum Jerk. La situation est illustrée en figure 3.6.

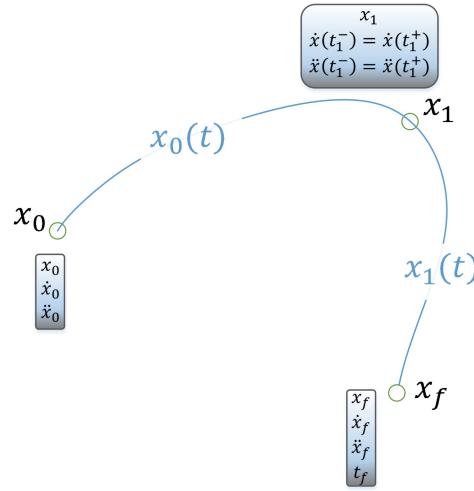


Figure 3.6 Illustration du problème de MJTO pour $n = 3$ points de passage.

On peut découpler le problème en deux problèmes distincts à résoudre itérativement : d'une part trouver les temps de passage optimaux selon le critère du MJ, d'autre part optimiser les coefficients des polynômes de $x(t)$ en supposant les t_i connus, comme on l'a fait à la section précédente (problème de MJTF). On va utiliser la programmation séquentielle convexe (*Sequential Convex Programming*, SCP (Boyd, 2000)) afin de résoudre cela, comme illustré en figure 3.7.

L'optimisation des coefficients des polynômes se fait relativement rapidement ; l'optimisation des temps de passage est un problème non linéaire avec contraintes d'inégalité. Il s'agit de trouver les t_i dans $\Delta t_i = t_i - t_{i-1}$ minimisant

$$J = \sum_{i=1}^{n-1} J(x_i(t))$$

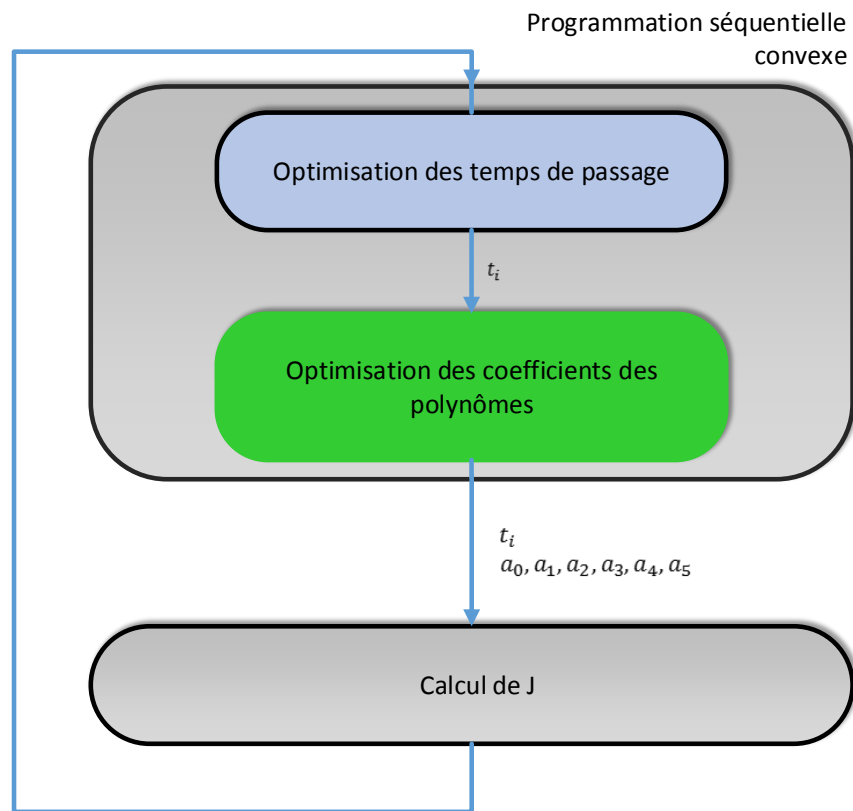


Figure 3.7 Principe de la programmation séquentielle convexe appliqué au problème de MJ avec temps de passage t_i optimisés.

$$\begin{aligned}
J(x_i(t)) &= \frac{1}{2} (36\Delta t_i a_{i3}^2 + 144\Delta t_i^2 a_{i3} a_{i4} + 192\Delta t_i^3 a_{i4}^2 + 240\Delta t_i^3 a_{i3} a_{i5} + \\
&\quad 720\Delta t_i^4 a_{i4} a_{i5} + 720\Delta t_i^5 a_{i5}^2) \\
&= a^\top Q(\Delta t_i) a.
\end{aligned}$$

sous les contraintes d'inégalités strictes

$$0 < t_1 < t_2 < \dots < t_f,$$

et avec les coefficients a_i déterminés en utilisant la SCP. Ce problème est plus complexe que le problème de MJTF avec t_i fixes et ne se fait pas en temps réel. On observe expérimentalement que l'algorithme converge vers un minimum, mais la forme du problème ne permet pas de déterminer s'il s'agit bien d'un minimum global. Le SCP est une méthode d'optimisation locale qui ne peut garantir un minimum global (Boyd, 2000).

3.4 Comparaison entre MJTF et MJTO

On observe un exemple de résultats pour une situation en 1D à la figure 3.8 pour les points de passage

$$P_x = [0, x_1, 1]^\top, \quad 0 < x_1 < 1,$$

où l'on a pris différentes valeurs pour x_1 . Les temps de passage initiaux sont ceux trouvés à l'Eq. 3.8. On observe que le temps t_i optimal est une fonction non linéaire du point de passage x_1 . Dans le cas symétrique où $x_1 = 0.5$ et avec les conditions initiales et finales $\dot{x}_0 = \dot{x}_f = \ddot{x}_0 = \ddot{x}_f = 0$, on a $t_1 = 0.5$.

Un exemple plus complexe de trajectoire 3D est illustré en figure 3.9. On observe que la trajectoire est différente selon que l'on optimise les temps de passage (MJTO) ou non (MJTF).

Plus particulièrement en figure 3.10, on voit l'influence des points de passage sur la différence entre le MJTO et le MJTF. On observe que pour des points de passage inégalement espacés, l'approximation des t_i de l'Eq. 3.8 est d'autant plus mauvaise et la différence avec le problème de MJTO est marquée. Dans l'exemple de la figure, les temps de passage intermédiaires initiaux (de MJTF) valent

$$t_{i0} = [0, 0.1618, 1.7151, 2]$$

et les temps optimisés (MJTO) valent

$$t_{i0} = [0, 0.4568, 1.4281, 2].$$

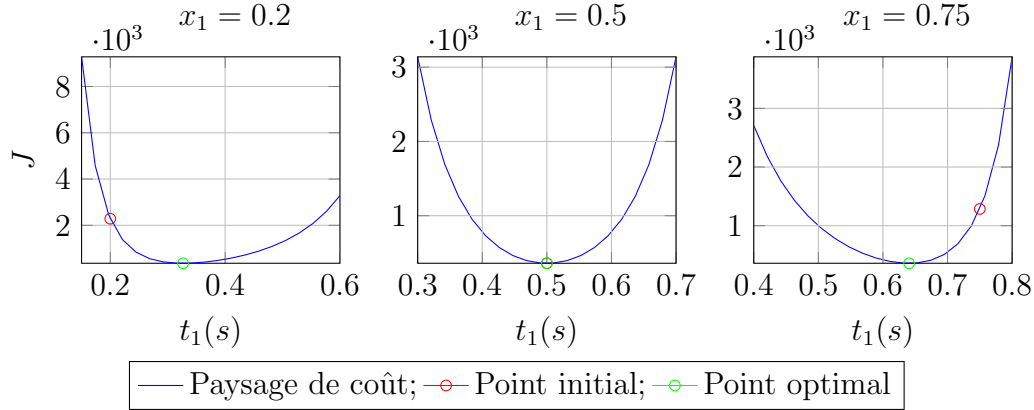


Figure 3.8 Optimisation de t_1 pour une trajectoire à $n = 3$ points de passage en 1D. $x_1 = 0.2$ (gauche), $x_1 = 0.5$ (milieu), $x_1 = 0.75$ (droite). Les temps de passage initiaux correspondent aux temps de passage pour le problème de MJTF.

On peut observer que le jerk pour le MJTO est plus faible que pour le MJTF. On voit donc l'importance de ne pas simplement imposer des temps de passage arbitraire, qui ne permet pas d'obtenir les trajectoires les plus naturelles.

3.5 Comparaison entre MJ et DL

On compare ici les trajectoires obtenues entre le modèle du MJ (MJTO) et celui du DL. On observe les résultats pour un cas 2D en figure 3.11. On observe que le modèle DL est capable de représenter l'asymétrie typique d'un mouvement naturel alors que le MJ garde une forme symétrique pour le profil de vitesse en cloche.

On compare maintenant une trajectoire plus complexe, l'écriture de trois lettres 'hms' en 2D. Les résultats sont illustrés en figure 3.12. La loi DL requiert 120 paramètres pour cette trajectoire (20 profils de vitesse DL) comparé à 100 paramètres (50 points de passage en 2 dimensions) pour le MJTO.

On observe que pour une trajectoire plus complexe, le MJ et DL sont tous les deux proches, bien que différents, sans doute de par l'asymétrie du profil de vitesse qui n'est pas capturé par le modèle de MJ. On observe également que plus on a de points de passage pour une même trajectoire, plus le problème s'apparente à une interpolation de points, l'optimisation n'ayant que peu d'impact sur l'apparence finale de la trajectoire.

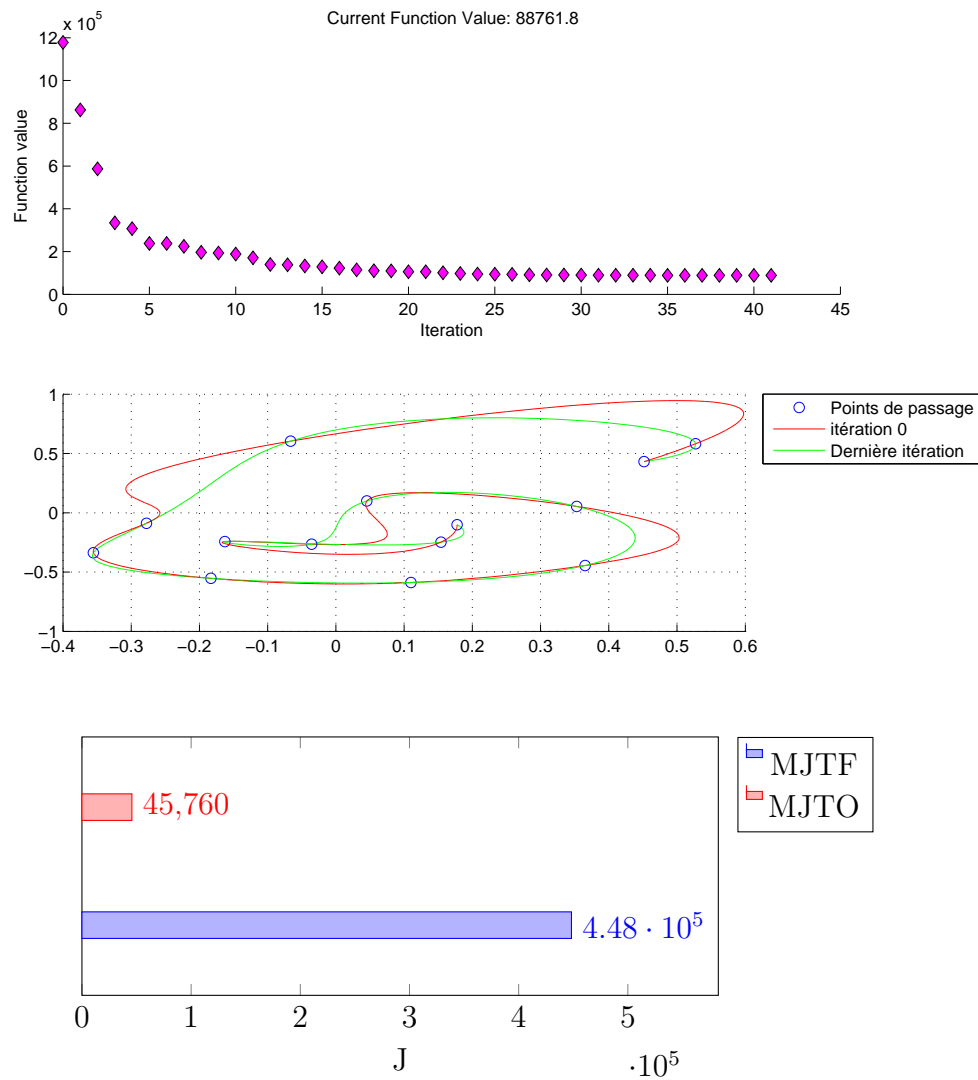


Figure 3.9 Comparaison de trajectoires de MJTF et MJTO. Fonction de coût au fur et à mesure de l'optimisation (haut), profil de positions (milieu), temps de calculs pour le MJTO et MJTF respectivement (bas).

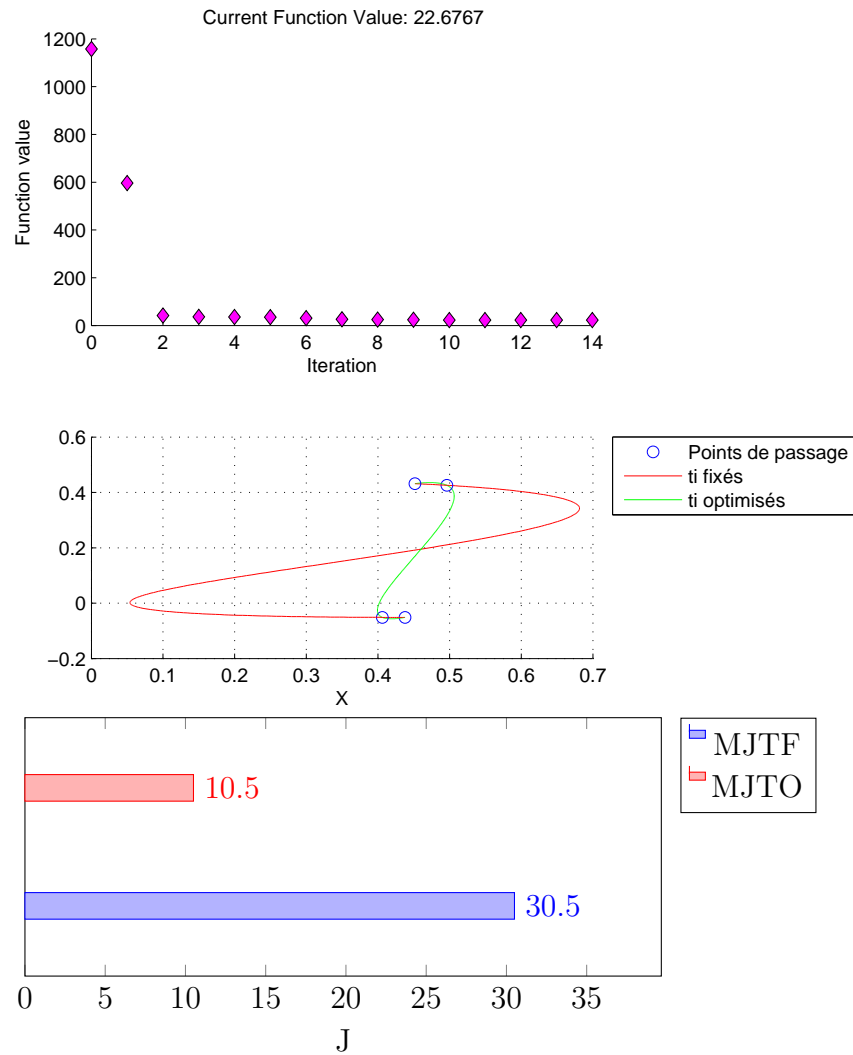


Figure 3.10 Comparaison entre l'optimisation de MJTF et MJTO. Évolution de la fonction de coût pour le MJTO en haut (l'itération 0 correspond au cas MJTF). Positions X-Y des deux trajectoires au milieu. Jerks des deux trajectoires en bas.

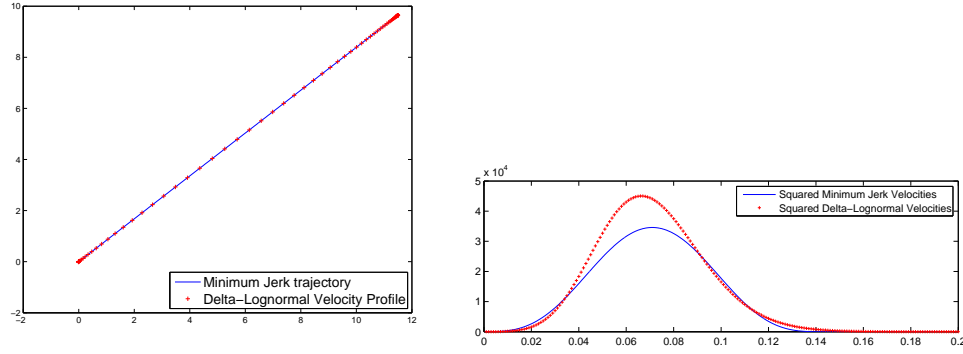


Figure 3.11 Comparaison d'une trajectoire point à point entre le modèle du MJ et celui du DL (gauche) ainsi que le profil de vitesse associé (droite).

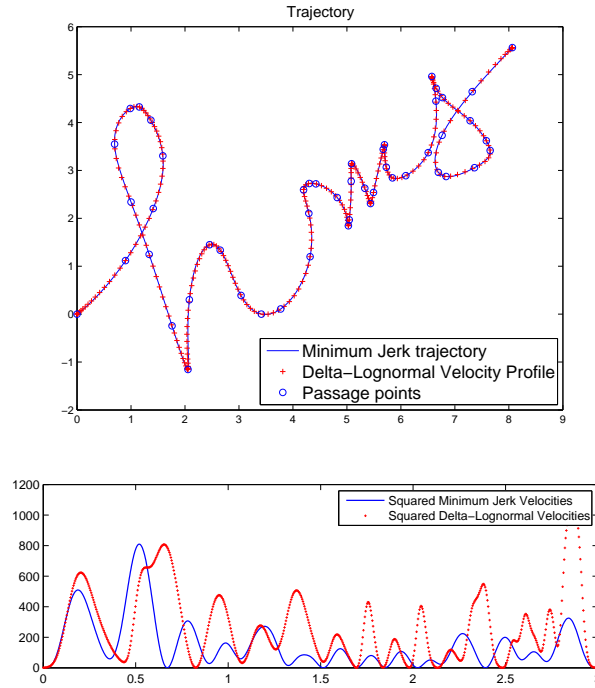


Figure 3.12 Comparaison entre MJ et DL pour l'écriture des trois lettres 'hms' en 2 dimensions. $n = 50$ points de passage pour le MJ. Positions à gauche, carré des vitesses à droite. Les résultats pour le DL sont obtenus en se basant sur la théorie Delta-Lognormal avec les paramètres décrits dans Plamondon and Djioa (2006).

3.6 Comparaison avec le blending

Le *blending*, ou animation paramétrique, est une technique largement utilisée en infographie qui permet de fondre deux mouvements entre eux afin de passer de l'un à l'autre de façon progressive (Kovar et al., 2002)(Heck and Gleicher, 2007). La technique utilise une interpolation du type

$$x_c(t) = \alpha(t)x_a(t) + \beta(t)x_b(t) \quad (3.17)$$

avec typiquement

$$\beta(t) = (1 - \alpha(t))$$

$$0 \leq \alpha(t) \leq 1$$

et

$$\alpha(t) = t$$

fonction linéaire du temps (Fig. 3.13).



Figure 3.13 Illustration du passage d'un mouvement de course à un mouvement de sprint en utilisant la technique du blending pour le jeu vidéo *Assassin's Creed* (Cooper, 2013)².

Le blending se fait en temps réel même pour des corps à grand nombre de DDL (typiquement 30 à 50). La technique est principalement utilisée afin d'assembler des mouvements indépendants d'un corps complexe. Par exemple, on peut mélanger une animation d'un personnage marchant et un personnage levant la main, afin d'obtenir une nouvelle animation du

personnage levant la main tout en marchant (Fig. 3.14).

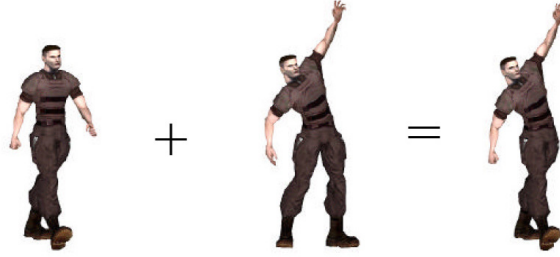


Figure 3.14 Illustration de l'utilisation du blending pour mélanger deux mouvements complémentaires. On mélange un mouvement de marche avec un mouvement de main levée (images issues du site *gamerendering*).

Le blending est utilisé de façon vaste dans le domaine de l'infographie principalement pour sa facilité d'utilisation, sa facilité de fonctionnement en temps réel et ses bons résultats. La technique ne satisfait en revanche pas de critère d'optimisation particulier afin de garantir la naturalité du mouvement obtenu. Al Borno et al. (2013) et Harada et al. (2006) ont prouvé qu'utiliser des méthodes de blending pour des mouvements de déplacement produit des comportements incohérents pour les déplacements de pieds au sol, où le pied du corps considéré se retrouve à glisser au sol plutôt que de prendre position ; l'utilisation de méthodes d'optimisation permet d'éviter cela en spécifiant des contraintes explicites du problème. Nous allons ici analyser la naturalité de trajectoires générées par blending en terme du critère du MJ.

3.6.1 Blending de deux trajectoires avec $\alpha(t)$, $\beta(t)$ linéaires

Prenons le cas typique du blending avec $\beta(t) = (1 - \alpha(t))$ et $\alpha(t) = \frac{t}{t_f} \in [0, 1]$, t_f étant la durée du mouvement. La trajectoire s'écrit alors

$$x_c(t) = \alpha(t)x_a(t) + \beta(t)x_b(t) = \left(\frac{t}{t_f}\right)x_a(t) + \left(1 - \frac{t}{t_f}\right)x_b(t).$$

On peut réécrire $x_c(t)$ sous la forme

$$x_c(t) = \mathbf{t}^\top \mathbf{a}_c$$

avec

$$\mathbf{t} = \begin{pmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 & t^6 \end{pmatrix}$$

et

$$\mathbf{a}_c = \begin{pmatrix} a_{0b} \\ \frac{a_{0a}}{t_f} + a_{1b} - \frac{a_{0b}}{t_f} \\ \frac{a_{1a}}{t_f} + a_{2b} - \frac{a_{1b}}{t_f} \\ \frac{a_{2a}}{t_f} + a_{3b} - \frac{a_{2b}}{t_f} \\ \frac{a_{3a}}{t_f} + a_{4b} - \frac{a_{3b}}{t_f} \\ \frac{a_{4a}}{t_f} + a_{5b} - \frac{a_{4b}}{t_f} \\ \frac{a_{5a}}{t_f} - \frac{a_{5b}}{t_f} \end{pmatrix}^\top$$

a_{ia} étant le coefficient i de $x_a(t)$ et a_{ib} étant le coefficient i de $x_b(t)$. En rappelant l'Eq. 3.3 de la section 3.1, on doit avoir $\forall t$

$$a_{6c} = \frac{a_{5a}}{t_f} - \frac{a_{5b}}{t_f} = 0$$

donc

$$a_{5a} = a_{5b}$$

ce qui n'est pas nécessairement le cas. Les autres coefficients ne remplissent pas les conditions nécessaires du problème non plus. Si nous avons par exemple

$$\dot{x}_a(t_0) = \dot{x}_b(t_0) = \ddot{x}_a(t_0) = \ddot{x}_b(t_0) = 0.$$

On doit également avoir

$$\dot{x}_c(t_0) = 0,$$

ce qui implique

$$a_{1c} = \frac{a_{0a}}{t_f} + a_{1b} - \frac{a_{0b}}{t_f} = \frac{a_{0a}}{t_f} - \frac{a_{0b}}{t_f}$$

Ce qui ne sera le cas que pour le cas particulier $x_a(t_0) = x_b(t_0)$. Tous les autres cas ne respectent donc pas le critère du MJ (MJTF et MJTO). On pourrait étendre le cas pour d'autres $\alpha(t)$ et $\beta(t)$ non constant, mais le cas typique étudié ici montre que le blending tel qu'utilisé habituellement ne donne pas de résultats naturels en terme du critère du MJ. On pourrait essayer de trouver sous quelles conditions sur la forme de $\alpha(t)$ et $\beta(t)$ la trajectoire résultante serait naturelle, mais ceci sort du cadre de ce mémoire.

Il est intéressant de noter ici l'intérêt d'analyser des trajectoires à $n > 2$ points de passages. Une trajectoire à $n = 2$ points de passage avec durée imposée, comme c'est classiquement

le cas dans la littérature, mélangée avec une autre trajectoire du même type donnera une nouvelle trajectoire qui sera elle-même le minimum du nouveau problème de MJ étant donné qu'aucun temps de passage ne doit être optimisé. Ceci n'est en revanche pas le cas pour des trajectoires de MJTO du fait que les temps de passages sont différents, et que le problème ne peut plus être représenté comme un système KKT augmenté linéaire. Ceci est corroboré par les nombreuses critiques du blending linéaire, qui est en général restreint pour des mouvements de courte durée et donc les paramètres sont évalués expérimentalement pour chaque utilisation.

On peut en conclure que le blending ne permet pas automatiquement de faire des transitions entre deux mouvements naturels. Cooper (2013) explique également que réaliser du blending d'animation par ordinateur consiste à trouver les moments, durées et forme du blending à la main pour chaque animation.

3.6.2 Blending de deux trajectoires avec $\alpha(t)$, $\beta(t)$ constants

Intéressons-nous maintenant à la formation de nouveaux mouvements à partir de mouvements naturels déjà existants. Prenons une trajectoire formée d'un mélange de deux trajectoires de MJTF. On va tout d'abord supposer $\beta(t) = (1 - \alpha(t))$ et $\alpha(t) = \alpha = \text{Cste}$ et ainsi réexprimer l'Eq. 3.17

$$x_c(t) = \alpha x_a(t) + (1 - \alpha)x_b(t).$$

Faire cela revient à pondérer de la même façon les coefficients a_i des trajectoires $x_a(t)$ et $x_b(t)$

$$a_c = \alpha a_a + \beta a_b. \quad (3.18)$$

On prends comme hypothèse pour le problème de MJ que les temps de passages t_i pour $x_a(t)$ et $x_b(t)$ sont identiques. Comme démontré en section 3.2, le problème de MJTF peut s'exprimer comme un problème augmenté linéaire. Il vient donc que la fonction de coût augmenté avec les contraintes d'égalité vaut

$$H_c = \frac{1}{2} a_c^\top Q a_c + \lambda_c^\top (E a_c + d_c)$$

On dérive par rapport à λ_c pour trouver le minimum. En utilisant l'Eq. 3.18, on en déduit

$$\begin{aligned} \frac{dH_c}{d\lambda} &= E(\alpha a_a + \beta a_b) - (\alpha d_a + \beta d_b) = 0 \\ &= \alpha \underbrace{(E a_a - d_a)}_{=0} + \beta \underbrace{(E a_b - d_b)}_{=0} = 0. \end{aligned}$$

Les trajectoires $x_a(t)$ et $x_b(t)$ respectant leurs contraintes respectives, chacun des deux termes de l'équation ci-dessus respecte les contraintes $\forall \alpha, \beta$. On dérive enfin par rapport à a_c pour trouver le minimum

$$\begin{aligned} \frac{dH_c}{da_c} &= a_c^\top Q + \lambda_c^\top E = 0 \\ &= (\alpha a_a^\top Q + \lambda_a^\top E) + (\beta a_b^\top Q + \lambda_b^\top E) = 0 \\ &= \frac{dH_a}{da_a} + \frac{dH_b}{da_b} = 0. \end{aligned}$$

Étant donné que chaque trajectoire $x_a(t)$ et $x_b(t)$ minimisent leur fonction de coût respective, on en conclut que $x_c(t)$ minimise également sa fonction de coût augmentée H_c .

Comparaison avec MJTF

On observe immédiatement qu'utiliser des coefficients $\alpha(t)$, $\beta(t)$ constants et en imposant l'hypothèse que les t_i sont identiques pour $x_a(t)$ et $x_b(t)$ revient à faire une interpolation des points de passages de $x_a(t)$ et $x_b(t)$

$$x_c(t_i) = \alpha x_a(t_i) + \beta x_b(t_i).$$

On a montré théoriquement à la section précédente que les résultats d'un tel mélange pour deux trajectoires naturelles donnent un résultat naturel pour le problème de MJTF. On peut vérifier le résultat expérimentalement en mesurant l'erreur

$$|x_c(t) - x_{MJTF}(t)|.$$

On observe l'erreur pour 25 trajectoires générées aléatoirement avec $n = 4$ points de passage. Les erreurs sont illustrées en figure 3.15. Les deux trajectoires sont systématiquement confondues et on observe bien que l'erreur est de l'ordre de l'erreur numérique.

Comparaison avec MJTO

Si on compare maintenant le blending au modèle MJTO, le problème ne peut plus être réécrit comme un système linéaire augmenté. La solution optimale n'est dès lors plus une pondération linéaire des deux trajectoires de base et

$$J^* \leq J(x_c(t)).$$

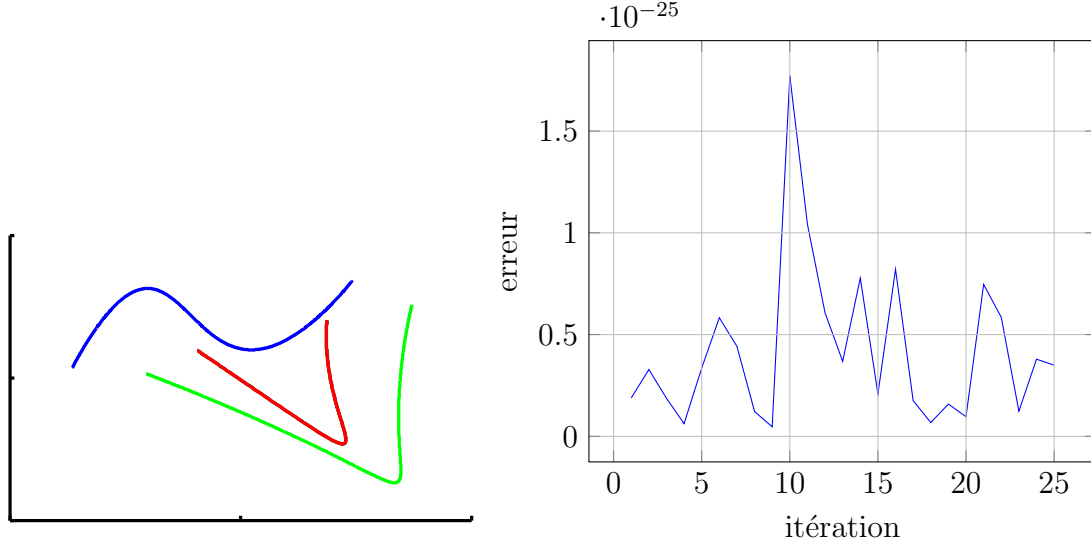


Figure 3.15 Comparaison entre le blending (à coefficients constants) entre deux trajectoires avec temps de passage identiques et le modèle de MJTF pour 25 trajectoires différentes à $n = 4$ points de passage. Une des trajectoires est illustrée à gauche (bleu et vert : trajectoires de base ; rouge : MJTO ; noir : blending), les erreurs pour chaque essai sont illustrées à droite. On observe que les résultats du MJTF et du blending sont identiques.

On vérifie cela en comparant le blending au modèle MJTO. Les résultats sont illustrés en figure 3.16. On observe ici que les trajectoires sont systématiquement différentes et que les erreurs ne sont ici plus négligeables. Ceci est dû au caractère non linéaire du problème de MJTO, pour lequel le système ne peut plus s'exprimer comme un système linéaire augmenté comme pour le MJTF.

Rappelons l'hypothèse initiale que nous avons prise que les temps de passage t_i sont identiques pour les deux trajectoires. Si de plus les deux trajectoires mélangées sont proches les unes des autres, on supposera que les temps de passage pour la trajectoire résultante sont proches des temps t_i originaux des deux trajectoires. En supposant les temps t_i identiques, on peut ramener le problème au cas du MJTF pour lequel on a prouvé précédemment que le mélange constant de trajectoires donne un résultat naturel.

Ceci permet de montrer que des travaux tels ceux de Kim and Park (2008) et Kwon and Park (2008) générant de nouveaux mouvements à partir de plusieurs reproductions différentes d'un même mouvement permet également d'obtenir un nouveau mouvement qui est naturel.

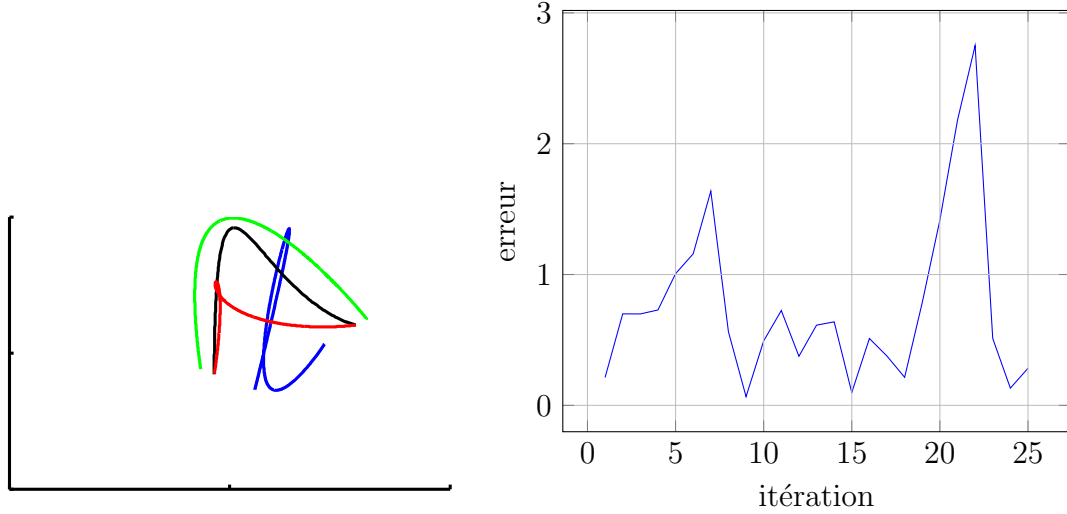


Figure 3.16 Comparaison entre le blending (à coefficients constants) entre deux trajectoires avec temps de passage optimisés et le modèle de MJTO. Une des trajectoires est illustrée à gauche (bleu et vert : trajectoires de base ; rouge : MJTO ; noir : blending), les erreurs pour chaque essai sont illustrés à droite. On observe que ceci est d'autant plus vrai que les t_i des mouvements de base sont différents.

3.7 Conclusion

Dans ce chapitre, nous avons étudié le problème du MJ sous différentes formes. Le problème pour $n = 2$ points de passage est un cas particulier se résolvant instantanément. Une forme simplifiée du problème, le MJTF, suppose les temps de passage imposés, peut être exprimée comme un problème d'optimisation quadratique avec contraintes d'égalité et peut se résoudre en temps réel. Le problème complet, le MJTO, optimise en plus les temps de passage afin de minimiser le jerk ; ce problème prend en revanche un certain nombre de temps et ne peut se résoudre en temps réel.

Enfin, nous avons comparé le problème de MJ entre deux points avec la technique de blending. Nous avons tout d'abord trouvé que le blending avec coefficients variables afin de créer des transitions entre mouvements ne minimise pas le critère de naturalité et ne permet pas instinctivement de générer des mouvements naturels. Nous avons ensuite également observé que mélanger deux mouvements naturels entre eux pour en créer un nouveau permet de générer un nouveau mouvement qui respectent le critère de naturalité du MJTF, sous l'hypothèse que les deux mouvements sont proches les uns des autres,

Le MJTF permet ainsi de générer des trajectoires en temps réel mais dont les temps de passage ne sont aucunement optimisés. Le MJTO, quant à lui, optimise en plus les temps de

passage pour une trajectoire à MJ minimum, mais prend du temps à trouver une solution. Nous étudierons au prochain chapitre une technique permettant de combiner les avantages des deux techniques.

CHAPITRE 4

Composition Séquentielle

Le MJTO présenté au chapitre précédent permet de générer des trajectoires naturelles selon le critère du MJ, mais est gourmand en temps de calcul. Nous développons ici une technique de composition séquentielle, permettant de tirer parti au maximum de calculs hors ligne MJTO effectués sur des mouvements de base, et de les utiliser afin de créer une trajectoire MJTF la plus naturelle possible, en un temps de calcul minimum.

4.1 Principe de la composition séquentielle

La composition séquentielle consiste à assembler des mouvements de base entre eux afin de générer rapidement de nouvelles trajectoires (Figs. 4.1 et 4.2).

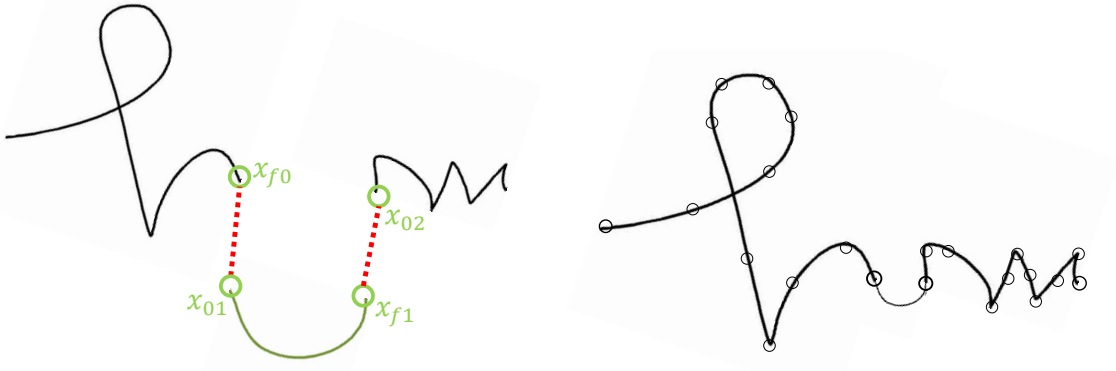


Figure 4.1 Principe de la composition séquentielle.

Les mouvements de base peuvent être obtenus selon différentes méthodes (Motion Capture ou MoCap, optimisation par MJ, etc.), le but étant d'obtenir un mouvement qu'on peut décrire comme naturel. Les mouvements sont reliés entre eux par des transitions. Afin que le mouvement global résultant paraisse naturel, il faut que ces transitions paraissent également naturelles.

On peut observer qu'une transition consiste en une trajectoire naturelle entre deux points de passage comme détaillé en section 3.1 ; la différence principale réside dans le fait que la

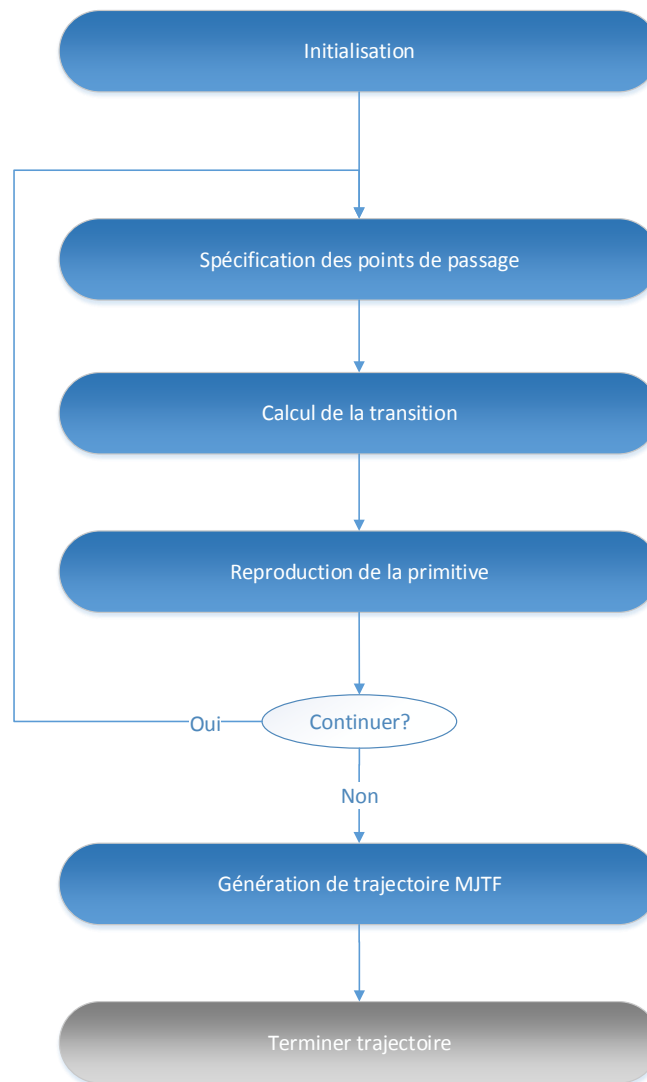


Figure 4.2 Pseudo-code de la composition séquentielle. La position de chaque primitive est déterminée par l'utilisateur.

durée de la transition est inconnue. Si celle-ci n'est pas imposée par l'utilisateur, elle peut être trouvée en utilisant la technique décrite en section 4.3.

Nous supposons ici que l'utilisateur impose la position où chaque primitive doit être reproduite, comme c'est le cas par exemple pour l'écriture manuscrite. La durée de transition d'une primitive à une autre peut soit être imposée arbitrairement (e.g., application avec mouvements cadencés à une vitesse imposée), soit être libre et optimisée afin d'obtenir un mouvement souple. C'est ce que nous allons tenter de faire à la section suivante. Nous verrons également comment interpréter la composition séquentielle en un nouveau problème de MJ avec temps de passage t_i imposés, afin d'obtenir une nouvelle trajectoire dont les mouvements repris des primitives de mouvement sont adaptées au mouvement complet ainsi généré.

Le problème avec un ensemble de points de primitives de mouvements et de transitions avec temps connus peut se traduire en un problème de MJTF comme détaillé en section 3.2. L'intérêt de considérer le problème comme un problème de MJTF plutôt qu'une trajectoire complètement imposée réside dans le fait que le problème de MJTF optimise les vitesses et accélérations à chaque point de passage ; on obtient donc une solution où la trajectoire de chaque primitive sera optimale pour les vitesses et accélérations aux points de passage pour le problème complet considéré, et non pas une simple reproduction de la primitive seule.

Il faut pour cela que l'on connaisse pour toute primitive les points et les temps de passages équivalents au problème de MJTO pour la primitive seule. Si la primitive a été obtenue par une autre technique (i.e., Mocap), on peut faire un fitting de cette MoCap par une trajectoire de MJ en extrayant des points de passage comme détaillé dans Wada and Kawato (2004). La technique (illustré en Fig. 4.3) ajoute itérativement comme point de passage l'élément de la trajectoire de référence le plus loin de la trajectoire reproduite. Comme expliqué en section 3.2, le problème posé de la sorte admet toujours une solution et cette solution est unique pour les contraintes imposées.

4.2 Optimisation de la trajectoire complète

En prenant les informations concernant les points de passage ainsi que les durées des transitions, on peut former un nouveau problème de MJTF afin de générer une nouvelle trajectoire dont les vitesses et les accélérations sont optimisées aux points de passage. On obtient ainsi une trajectoire dont l'apparence globale ne dépend pas uniquement des informations locales des primitives mais bien de l'ensemble de la trajectoire. En effet, le début et la fin des primitives seront visiblement influencés par le comportement de la trajectoire directement avant

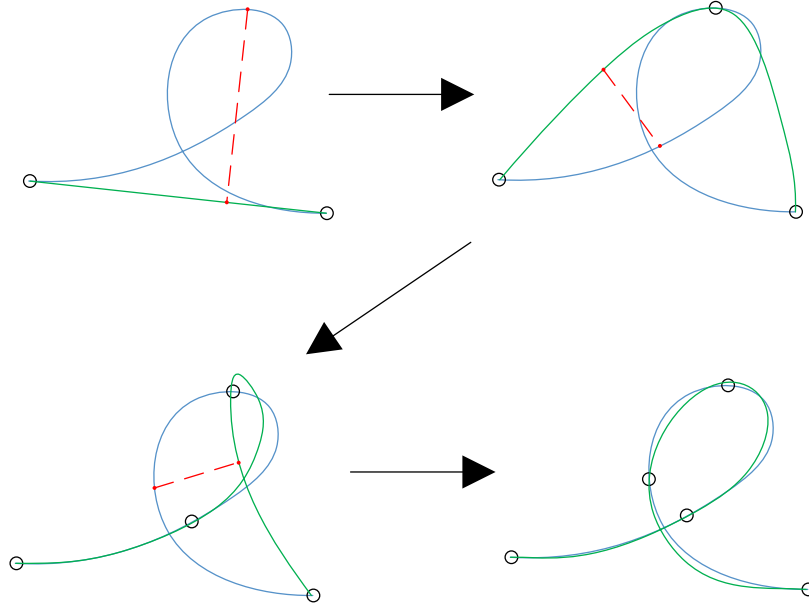


Figure 4.3 Illustration de la récupération de points de passage selon Wada and Kawato (2004).

et après. La théorie concernant les trajectoires de MJTF ont déjà été discutées en section 3.2. Si la durée des transitions est imposée, le calcul de MJTF est direct. Si les durées ne sont pas imposées, il faut les optimiser selon un critère développé ci-dessous.

4.3 Transition entre primitives de mouvements

Afin de décrire un critère exprimant la durée des transitions entre primitives, on désire minimiser l'écart entre les vitesses durant la transition et les vitesses des deux primitives adjacentes. La moyenne des vitesses sur la moitié de chaque primitive afin de définir la vitesse moyenne désirée pour la transition est donnée par :

$$\bar{v} = \frac{\int_{t_{fp1}/2}^{t_{fp1}} \dot{x}(t) dt + \int_0^{t_{fp2}/2} \dot{x}(t) dt}{(t_{fp1} + t_{fp2})/2}. \quad (4.1)$$

La fonction de coût adimensionnée vaut quant à elle

$$J_V(t_f) = \frac{\frac{1}{2} \int_{t_0}^{t_f} |\dot{x}(t) - \bar{v}|^2 dt}{\bar{v}^2}.$$

La situation est illustrée en figure 4.4.

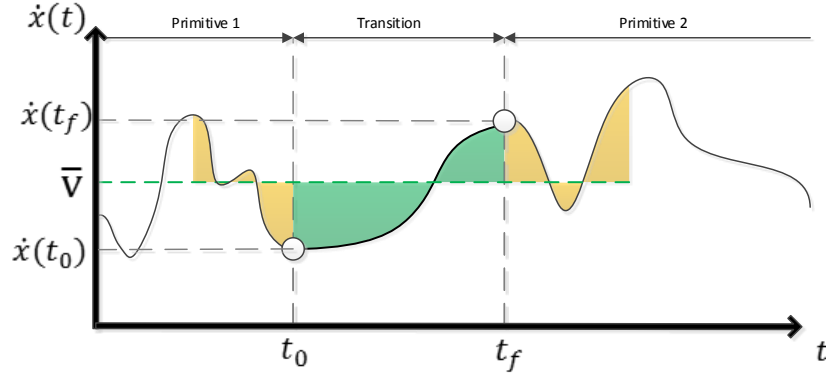


Figure 4.4 Illustration de la fonction de coût J_V et de la vitesse moyenne \bar{v} .

Nous observons qu'un tel critère donne de bons résultats, mais les durées des transitions obtenues sont trop longues d'un facteur constant proportionnel à la durée de la transition. Nous ajoutons donc un second critère à la fonction de coût de la forme

$$J_{t_f} = t_f - t_0,$$

où les deux critères sont pondérés par un facteur α trouvé expérimentalement en comparant le t_f obtenu avec celui obtenu avec la technique du MJTO. Le critère d'optimisation complet pour les transitions vaut donc :

$$\begin{aligned} J &= \alpha J_V(t_f) + (1 - \alpha) J_{t_f}(t_f) \\ J_V &= \frac{\frac{1}{2} \int_{t_0}^{t_f} |\dot{x}(t) - \bar{v}|^2 dt}{\bar{v}^2} \\ J_{t_f} &= t_f - t_0 \\ \bar{v} &= \frac{\int_{t_{fp1}/2}^{t_{fp1}} \dot{x}(t) dt + \int_0^{t_{fp2}/2} \dot{x}(t) dt}{(t_{fp1} + t_{fp2})/2}. \end{aligned} \tag{4.2}$$

On prend comme première approximation pour la durée de la transition

$$t_{f0} = \frac{\Delta x}{\Delta v} = \frac{|x_f - x_0|}{\left(\frac{|\dot{x}(0) + \dot{x}(t_f)|}{2} \right)} \tag{4.3}$$

En comparant les résultats de la composition séquentielle avec des trajectoires optimisées en

utilisant le MJ(MJTO), on trouve expérimentalement la valeur de $\alpha = 0.98$. Les deux critères J_V et J_{t_f} sont illustrés en figure 4.5.

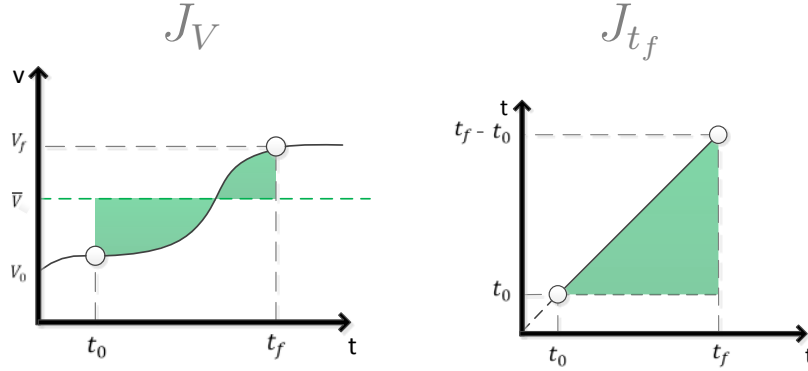


Figure 4.5 Forme des composantes de la fonction de coût des transitions.

Un exemple typique pour $\alpha = 0.95$ est illustré en figure 4.6 en partant du repos pour entamer une primitive obtenue par MoCap et reconvertie en un problème de MJ. La transition obtenue est illustrée en figure 4.7. Les temps de calcul sont de l'ordre de 1s à 5s par transition sur Matlab pour un ordinateur avec un processeur Intel i7-4770@3.4GHz et 12Go de RAM.

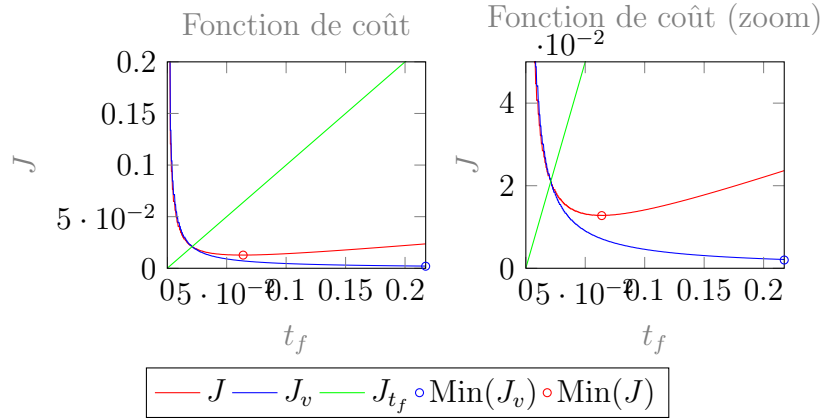


Figure 4.6 Exemple typique de la fonction de coût pour $\alpha = 0.95$.

4.4 Résultats

Observons maintenant les résultats d'un tel système. Un premier exemple est illustré en figure 4.8, et les temps de calcul correspondants sont illustrés en figure 4.10. Comme vu en section 3.2, le problème de MJTF se résout instantanément. L'optimisation des temps de

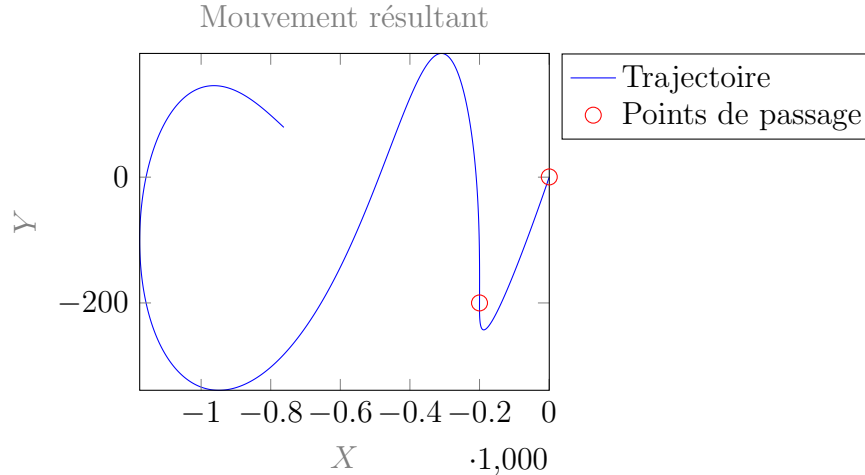


Figure 4.7 Mouvement résultant pour la fonction de coût de la figure 4.6.

transitions prend un peu de temps et ne permet pas le temps réel ; les temps de calcul sont cependant inférieurs au problème de MJTO. Si on impose les temps de passage pour les transitions, on observe que le problème est solvable en temps réel. On peut voir que chacun des problèmes illustrés ont des transitions différentes : l'assemblage simple de primitive (sans optimisation ultérieure) a les transitions les moins naturelles à première vue, et chaque primitive est identique exactement. Pour la composition séquentielle (assemblage des points de passage + optimisation) sans optimisation de la durée des transitions, on utilise directement l'approximation de l'Eq. 4.3. On observe que cette solution donne les résultats les plus proches avec ceux du MJTO. La composition séquentielle avec optimisation des transitions donne un résultat intermédiaire entre celles de l'assemblage pur et du MJTO.

Un autre exemple est illustré en figure 4.11 ; les temps de calcul sont illustrés en figure 4.13. Ici, on fait les mêmes observations pour l'assemblage pur de primitives. La transition entre les deux mouvements représentant le signe infini pour la composition séquentielle avec optimisation des transitions est cette fois-ci plus proche du MJTO que sans optimisation des transitions. Les différences entre la composition séquentielle avec ou sans optimisation des temps de passage dépendent du cas.

L'avantage de résoudre un problème de MJ avec temps fixés plutôt que d'assembler directement les trajectoires réside dans le fait qu'on va obtenir des vitesses et accélérations aux points de passages qui sont optimisés pour chaque situation.

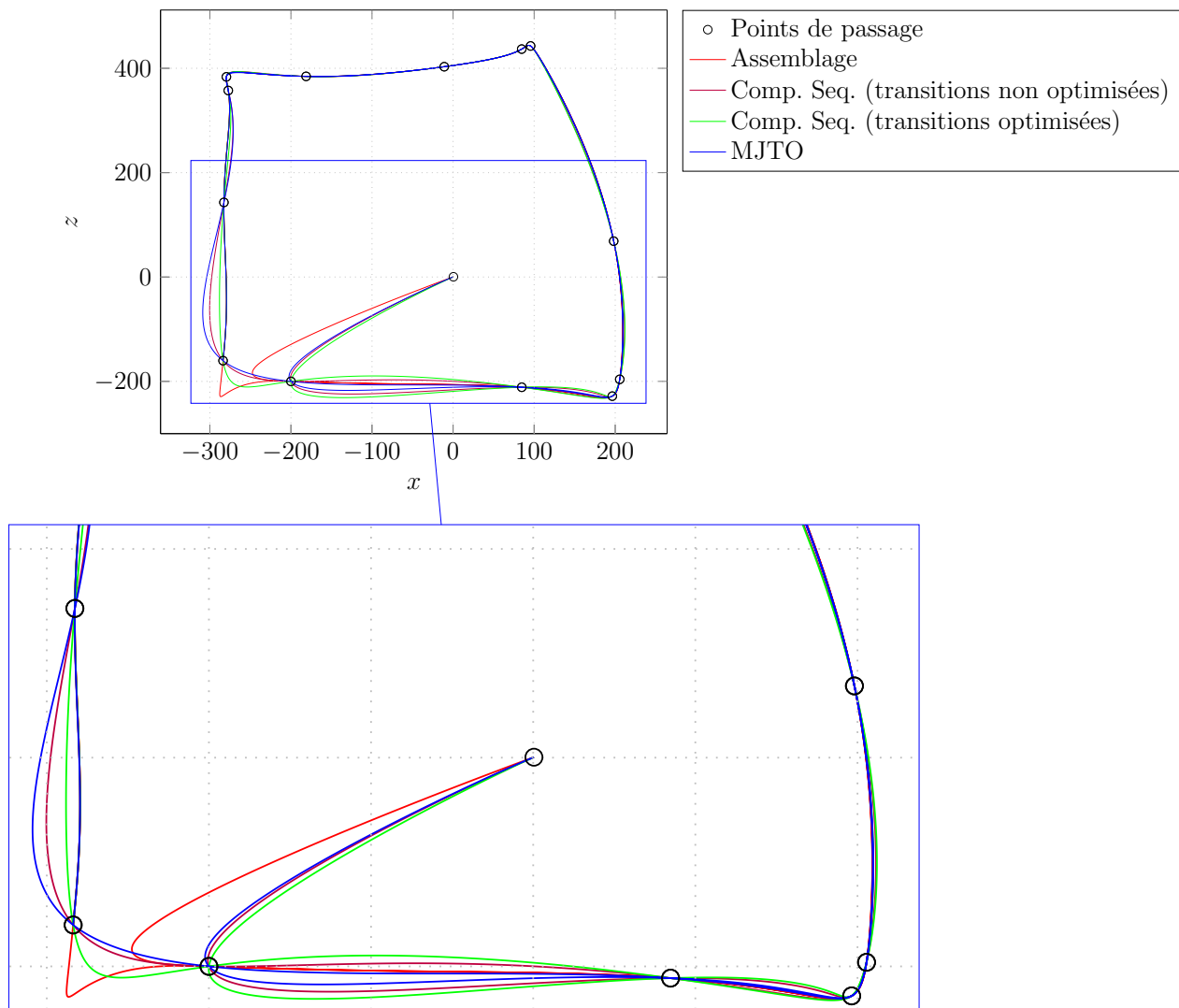


Figure 4.8 Exemple de trajectoire générée en utilisant le MJTO, la composition séquentielle (avec et sans optimisation des transitions), et en assemblant simplement les trajectoires des primitives ensemble. Trajectoire générale en haut, zoom sur les transitions en bas.

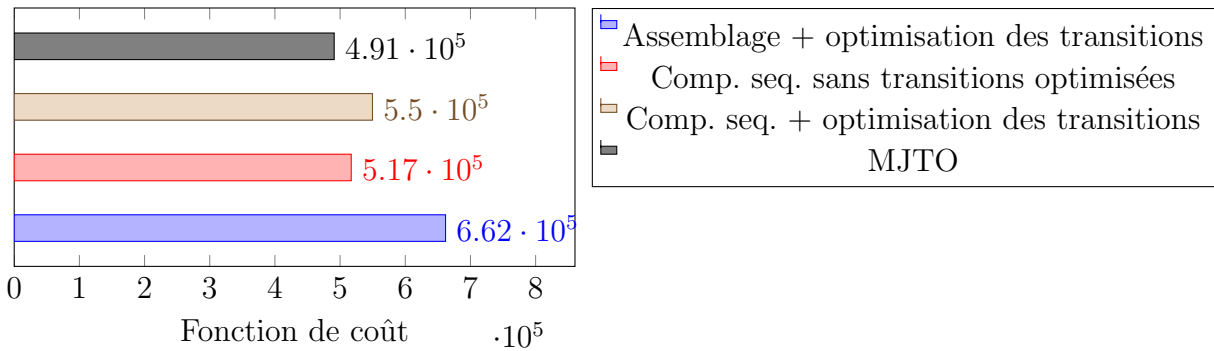


Figure 4.9 Intégrale de la norme du jerk pour l'exemple de la figure 4.8 pour différentes techniques.

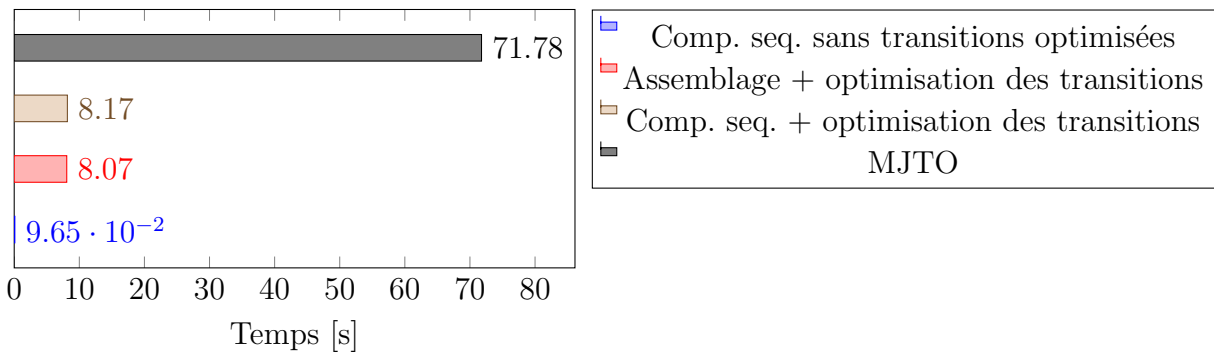


Figure 4.10 Temps de calcul pour l'exemple de la figure 4.8 pour différentes techniques. Les temps ont été obtenus avec la configuration décrite en section 4.3.

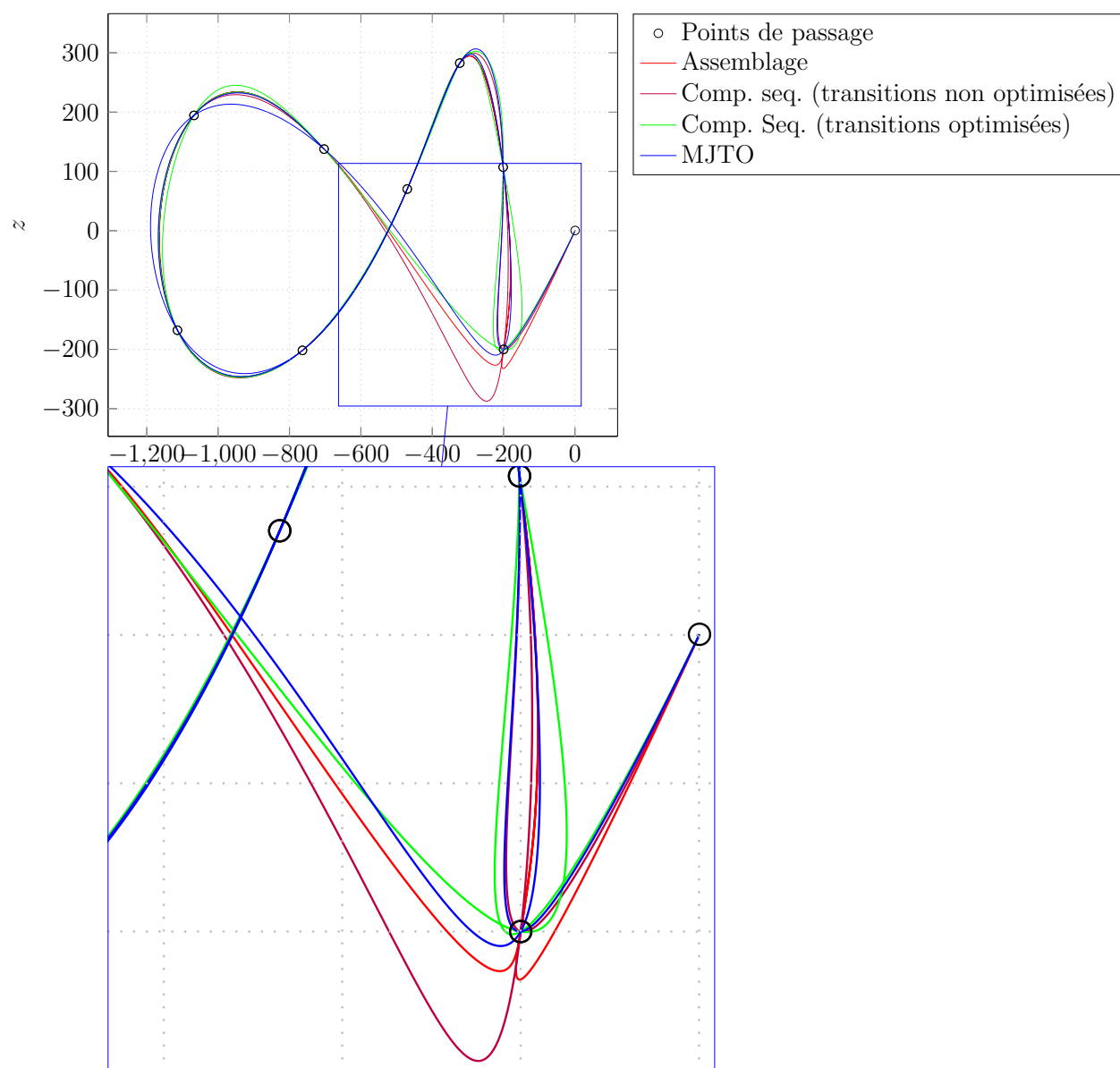


Figure 4.11 Autre exemple de trajectoire générée en utilisant le MJTO, la composition séquentielle (avec et sans optimisation des transitions), et en assemblant simplement les trajectoires des primitives ensemble. Trajectoire générale en haut, zoom sur les transitions en bas.

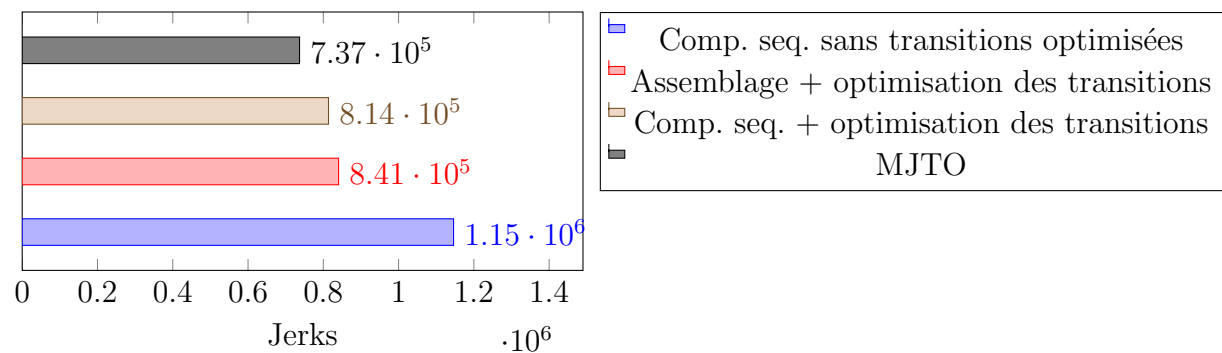


Figure 4.12 Intégrale de la norme du jerk pour l'exemple de la figure 4.8 pour différentes techniques.

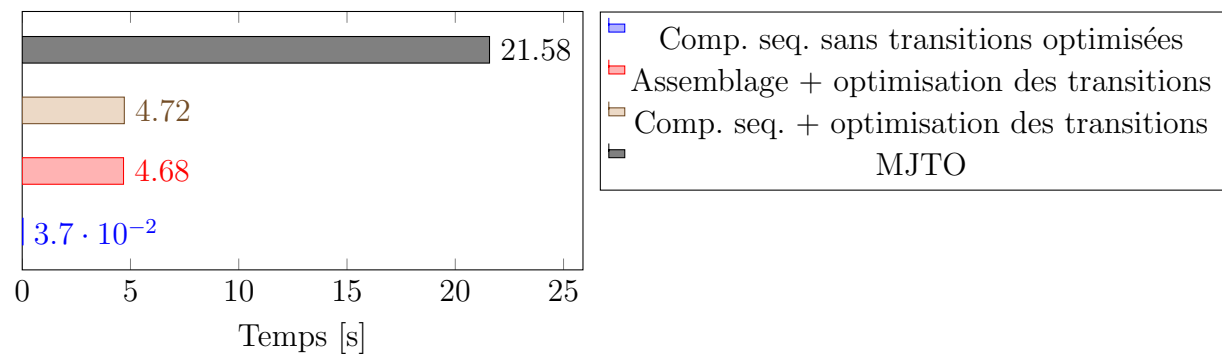


Figure 4.13 Temps de calcul pour l'exemple de la figure 4.11 pour différentes techniques. Les temps ont été obtenus avec la configuration décrite en section 4.3.

4.5 Timescaling

Il est intéressant de savoir dans quelles conditions on peut transformer une trajectoire originale $x(t)$ en une nouvelle trajectoire tout en gardant la naturalité de la trajectoire. On peut par exemple changer la durée de la trajectoire et observer l'impact sur son critère de naturalité, ici le jerk. Soit une trajectoire $x_{t_f}(t)$ d'une durée t_f . On veut mettre à l'échelle la durée du mouvement pour une durée $t_{f'}$. En effectuant le changement de variable

$$t \rightarrow \left(\frac{t_{f'}}{t_f} \right) t$$

On obtient pour l'expression du jerk

$$\frac{d^3 x(t)}{d \left[\left(\frac{t_{f'}}{t_f} \right) t \right]^3} = \left(\frac{t_f}{t_{f'}} \right)^3 \frac{d^3 x(t)}{dt^3}$$

L'ensemble du paysage de coût MJ est donc modifié par un terme constant. La position des minimums ne change pas, donc si $x_{t_f}(t)$ est bien minimum pour une durée de trajectoire t_f , $x_{t_{f'}}(t)$ est le minimum pour une trajectoire de durée $t_{f'}$.

En pratique, la trajectoire sera modifiée selon

$$x_{t_{f'}}^{(n)}(t) = \left(\frac{t_f}{t_{f'}} \right)^n x_{t_f}^{(n)}(t),$$

n correspondant à l'ordre de la dérivée.

On peut ainsi accélérer ou ralentir toute trajectoire planifiée à la volée tout en gardant la naturalité de la trajectoire.

4.6 Conclusion

Nous avons présenté un système de composition séquentielle basé sur l'assemblage de plusieurs mouvements individuels entre eux et l'optimisation du mouvement global. Toutes les primitives sont obtenues hors-ligne et exprimées comme un problème de MJ. Les contraintes de chaque primitive sont assemblées afin de former un problème de MJTF unique pour toute la trajectoire. La durée des transitions entre les primitives peut soit être optimisée ou imposée. Si elles sont optimisées, le problème se résout en un temps qui augmente linéairement avec le nombre de transitions ; la durée de résolution est plus faible que celle du MJTO. Si les

durées des transitions sont imposées, la résolution du problème global se fait en temps réel.

Les techniques présentées permettent ainsi de répondre au besoin de génération de trajectoire naturelle interactive : en fixant la durée des transitions, on est capable de générer immédiatement des trajectoires à l'apparence naturelle composée de mouvements primitifs. Dans le cas où la durée des transitions n'est pas imposée, on peut réaliser une optimisation plus rapide que pour le cas du MJTO. Le temps requis pour une telle optimisation est typiquement de 2 secondes par transition, ce qui remplit bien l'objectif de recherche précisé en début de ce mémoire, qui est de générer une trajectoire en moins de 10 secondes pour 5 primitives de mouvement.

Les trajectoires générées peuvent de plus être accélérées ou ralenties en temps réel sans perte de naturalité.

CHAPITRE 5

Prise en considération du domaine articulaire

Dans les chapitres précédents, nous avons vu comment générer une trajectoire naturelle dans le domaine cartésien, c-a-d une trajectoire représentant le bout de l'effecteur d'un robot. Les mêmes techniques peuvent s'appliquer dans le domaine cartésien ou articulaire, c-a-d dans le domaine des articulations du robot, mais pas les deux. Une trajectoire naturelle dans le domaine cartésien n'est ainsi pas forcément naturelle dans le domaine articulaire, et une trajectoire naturelle dans le domaine articulaire n'est pas forcément naturelle dans le domaine cartésien. Nous présentons ici une méthode permettant d'optimiser la naturalité de la trajectoire dans les deux domaines.

Ce chapitre décrit tout d'abord les différents aspects permettant de générer des trajectoires dans le domaine articulaire du robot en partant d'une trajectoire du domaine cartésien. Nous allons tirer parti de plusieurs algorithmes de cinématique inverse (CI) différents afin de bénéficier d'algorithmes d'évitement de singularité tout en minimisant les temps de calcul. Nous verrons également une technique afin de générer une trajectoire de MJ hybride entre les domaines articulaire et cartésien.

La CI analytique avec découplage du problème d'orientation et de positionnement pour deux robots industriels sont expliquées à la section 5.1, et le principe de la CI itérative est présenté en section 5.2 avec le *Resolved Motion Rate*, afin de pallier les problèmes de la première technique. Une utilisation des deux techniques en alternance est présentée en section 5.3 afin de permettre un évitement des singularités avec une charge computationnelle réduite. Une technique d'optimisation mixte du jerk dans les domaines cartésien et articulaire est enfin présentée en section 5.3.

5.1 Cinématique inverse analytique

Les résultats dans cette section permettent de réaliser la CI analytique pour deux robots différents.

5.1.1 FANUC LR Mate 200iC

Le LR Mate 20iC est un robot manipulateur à 6 DDL de FANUC Robotics. Il est illustré en figure 5.1. Les calculs de la CI analytique pour ce robot sont décrits en détail en annexe A.2. L'ensemble des solutions pour la CI du FANUC LR Mate 200iC est résumée dans le tableau 5.1 à titre indicatif. Les détails de notation sont décrits dans l'annexe.

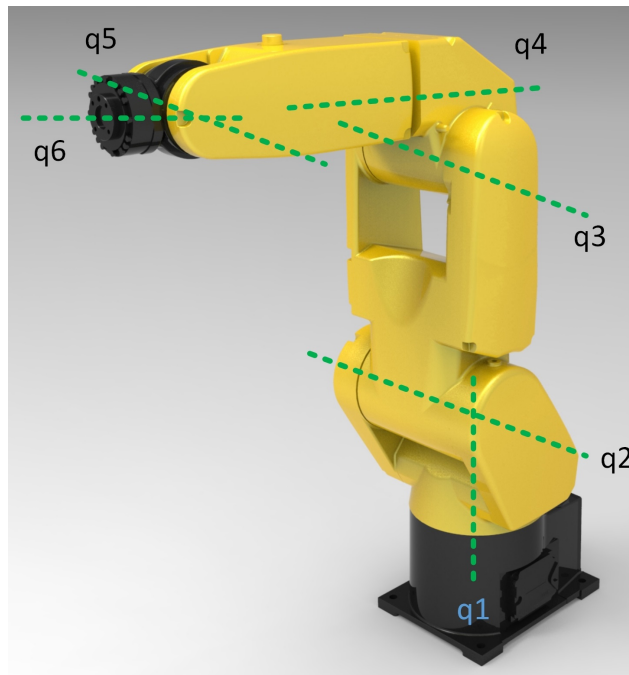


Figure 5.1 FANUC LR Mate 200iC.

5.1.2 ABB IRB 120T

Le robot IRB 120T est un robot manipulateur à 6 ddl de la compagnie ABB. Sa structure géométrique est très similaire à celle du FANUC LR Mate 200iC. Le robot est illustré en figure 5.2. Les détails des calculs de la CI analytique sont décrits en annexe A.3. L'ensemble des solutions pour la CI du ABB IRB 120T est résumée dans le tableau 5.2 à titre indicatif. Les détails de notation sont décrits dans l'annexe.

5.1.3 Choix de la solution analytique

La CI analytique admet plusieurs solutions. Pour un manipulateur à 6 DDL, nous avons 8 solutions possibles. Voici deux critères pour choisir une de ces solutions parmi les autres ; le

Tableau 5.1 Solution de la CI du FANUC LR Mate 200iC. 8 solutions distinctes sont possibles.

valeur	# solutions
$q_1 = \text{atan2}(\pm p_y, \pm p_x)$	2
$q_2 = \text{atan2}(M, L) - \text{atan2}(N, H)$	1
$q_3 = \text{atan2}(\pm \sqrt{1 - \left(\frac{A}{\rho}\right)^2}, \frac{A}{\rho}) - \text{atan2}(d_4, a_3)$	2
$q_4 = \begin{cases} \text{atan2}(a_y, a_x) \\ \text{atan2}(-a_y, -a_x) \end{cases}$	2
$q_5 = \text{atan2}(C_4 a_x + S_4 a_y, -a_z)$	1
$q_6 = \text{atan2}(S_4 n_x - C_4 n_y, S_4 o_x - C_4 o_y)$	
$A = \frac{p_x^2 + p_y^2 + (d_1 - p_z)^2 + a_I^3 \times 3 \cdot 2 - a_2^2 - a_3^2 - d_4^2 - 2a \sqrt{p_x^2 + p_y^2}}{2a_2}$ $\rho = \sqrt{a_3^2 + d_4^2}$ $H = C_1 p_x + S_1 p_y - a_1$ $L = -C_3 - a_3 S_3$ $M = a_3 C_3 - S_3 d_4 + a_2$ $N = p_z - d_1$ $Q = {}^0 R_3^T {}^0 R_6 R_x(\pi)$	

Tableau 5.2 Solution de la CI du ABB IRB 120T. 8 solutions distinctes sont possibles.

valeur	# solutions
$q_1 = \text{atan2}(\pm p_y, \pm p_x)$	2
$q_2 = \text{atan2}(M, L') - \text{atan2}(N, H)$	1
$q_3 = \text{atan2}(\pm \sqrt{1 - \left(\frac{A}{\rho}\right)^2}, \frac{A}{\rho}) - \text{atan2}(d_4, a_3)$	2
$q_4 = \begin{cases} \text{atan2}(a_y, a_x) \\ \text{atan2}(-a_y, -a_x) \end{cases}$	2
$q_5 = \text{atan2}(C_4 a_x + S_4 a_y, a_z)$	1
$q_6 = \text{atan2}(-S_4 n_x + C_4 n_y, -S_4 o_x + C_4 o_y)$	1
$A = \frac{p_x^2 + p_y^2 + (d_1 - p_z)^2 + a_I^3 \times 3 \cdot 2 - a_2^2 - a_3^2 - d_4^2 - 2a \sqrt{p_x^2 + p_y^2}}{2a_2}$ $\rho = \sqrt{a_3^2 + d_4^2}$ $H = C_1 p_x + S_1 p_y - a_1$ $L' = C_3 + a_3 S_3$ $M = a_3 C_3 - S_3 d_4 + a_2$ $N = p_z - d_1$ $Q = {}^0 R_3^T {}^0 R_6 R_z(\pi)$	

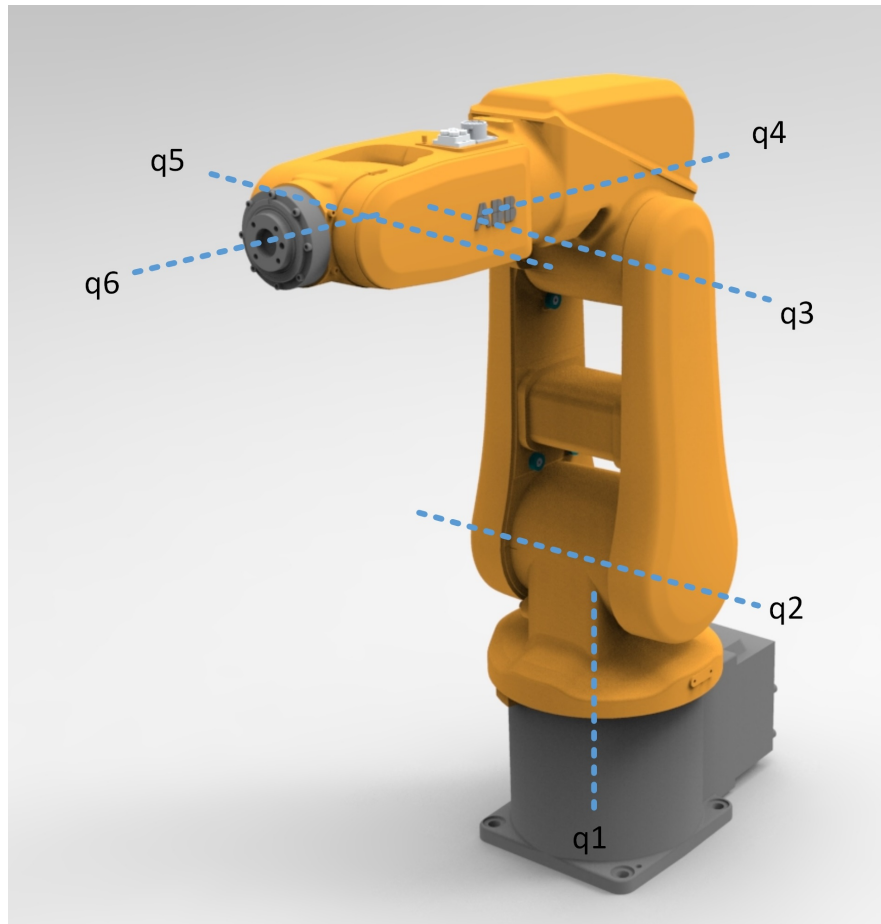


Figure 5.2 ABB IRB 120T.

premier pouvant s'exécuter en temps réel, le second nécessitant une analyse complète de la trajectoire.

En ligne - Choix à chaque pas de temps

Une première solution est de choisir la solution la plus proche de la position articulaire précédente respectant les limites articulaires. Cette technique a l'avantage d'être simple, calculable rapidement et à chaque pas de temps. Après calcul des 8 solutions analytique $q_{(1)}(t_i) \dots q_{(8)}(t_i)$, on vérifie pour chacune d'elles si chaque limite articulaire est respectée, c-a-d.

$$q_{min} \leq q \leq q_{max}. \quad (5.1)$$

Et pour celles respectant les limites, on choisit la solution minimisant

$$|q(t_i) - q(t_{i-1})|. \quad (5.2)$$

L'équation ci-dessus a du être adaptée au cas du dernier axe ($j = 6$) des deux robots analysés car sa limite articulaire s'étend au-delà de 2π . L'Eq. 5.2 devient pour l'articulation 6.

$$|q(t_i) - q(t_{i-1})| \mod 2\pi,$$

et on rajoute éventuellement $\pm 2\pi$ à $q(t_{i-1})$. Les limites articulaires des deux robots étudiés sont reprises dans le tableau 5.3.

Tableau 5.3 Limites articulaires des robots FANUC LR Mate 200iC et ABB IRB 120T.

Joint	FANUC LR Mate 200iC		ABB IRB 120T	
	$q_{j\ min}[^{\circ}]$	$q_{j\ max}[^{\circ}]$	$q_{j\ min}[^{\circ}]$	$q_{j\ max}[^{\circ}]$
1	-170	170	-165	165
2	-60	140	-110	110
3	-71.4	184.5	-90	70
4	-190	190	-160	160
5	-120	120	-120	120
6	-360	360	-400	400

Un tel choix de solution pose problème quand la configuration actuelle du robot dépasse une limite articulaire. En effet, deux choix s'offre à nous : saturer le résultat de la configuration actuelle, auquel cas le résultat de la CI ne sera pas correct avant que le robot revienne

à des valeurs articulaires correctes, soit changer de configuration, auquel cas le robot sera temporairement dans une position incorrecte, le temps de passer de l'ancienne à la nouvelle configuration. Après expérimentation, nous choisissons de garder la même configuration et de saturer les joints à leur limite articulaire ; en effet, on observe la plupart du temps que le robot dépasse ses limites articulaires pour un faible temps seulement, avant de revenir à l'intérieur de ses limites articulaires. Il est plus avantageux dans ce cas de saturer les articulations à leur limite et attendre que le robot reviennent dans une position convenable.

La valeur de $q(t_{i-1})$ est pris de la position articulaire $q_{free}(t_{i-1})$ non saturée par les limites articulaires comme illustré en figure 5.3. Cela permet de ne pas changer de configuration une fois qu'on a atteint les limites articulaires.

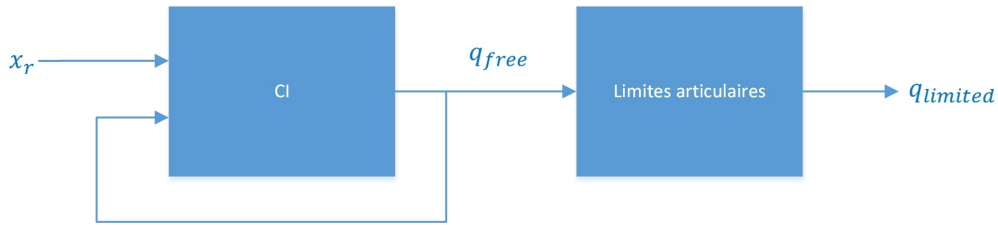


Figure 5.3 $q(t_{i-1})$ est pris sans limites articulaires.

Hors ligne - Algorithme de sélection de la trajectoire

On peut tirer parti du fait qu'on peut choisir la configuration initiale du robot afin de pallier (au moins en partie) au problème cité à la section précédente et suivre la trajectoire cartésienne désirée. Cette section décrit une technique hors-ligne permettant de choisir la configuration initiale permettant de réduire le problème de dépassements des limites articulaires.

On génère d'abord pour chacune des 8 configurations possibles, la trajectoire sans limites articulaires $q_{(i)free}(t)$, i étant l'indice de la solution. On applique ensuite les limites articulaires à chacune de ces trajectoires afin d'obtenir les $q_{(i)limited}(t)$. On peut ainsi calculer l'erreur totale obtenue après application des limites articulaires sur la trajectoire i :

$$e_i = \frac{1}{2} \int_{t_0}^{t_f} |q_{(i)free}(t) - q_{(i)limited}(t)| dt \quad (5.3)$$

On choisit enfin la trajectoire $q_{(i)limited}(t)$ pour laquelle l'erreur est minimum

$$q(t) = q_{(i)limited}(t), \quad i = \operatorname{argmin}(e)$$

qui est la trajectoire la plus proche de la trajectoire en domaine cartésien $x(t)$ initiale.

Un exemple de trajectoire est donné à la figure 5.4, et ses résultats sont illustrés en figure 5.5. Toutes ces trajectoires ont donc une erreur e_i plus grande que la trajectoire 3, qui est la trajectoire avec l'erreur la plus petite (Tab. 5.4). Nous prendrons donc ici les conditions initiales de la trajectoire 3, qui nous permettent d'avoir une trajectoire la plus proche possible de la trajectoire cartésienne de référence après application des limites articulaires.

Tableau 5.4 Évaluation de l'erreur de l'Eq. 5.3 pour l'exemple de la figure 5.4.

solution	e_i
1	0.0626×10^3
2	0.0626×10^3
3	0.0063×10^3
4	0.0063×10^3
5	3.1838×10^3
6	3.1838×10^3
7	0.9908×10^3
8	0.9908×10^3

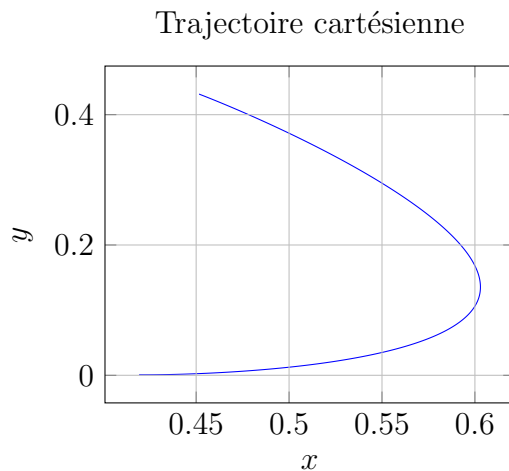


Figure 5.4 Trajectoire pour l'algorithme hors-ligne de sélection de trajectoires.

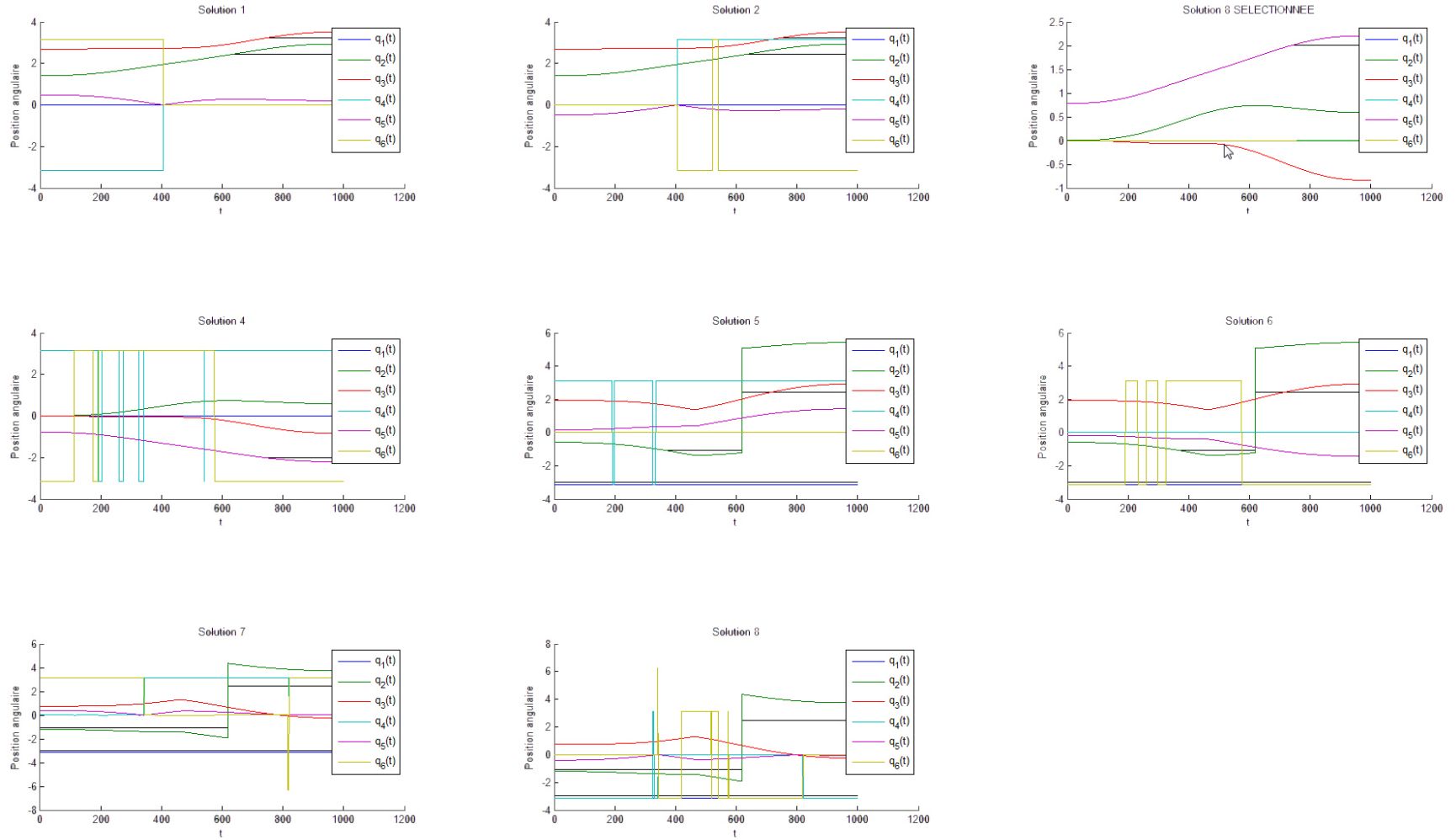


Figure 5.5 Choix de la condition initiale parmi les 8 solutions analytiques de la trajectoire de la figure 5.4. Les positions articulaires en couleur sont sans limites articulaires. Les positions articulaires avec application des limites sont illustrées en noir. La solution idéale choisie est ici la solution 3 (en haut à gauche).

5.2 Cinématique inverse itérative : Resolved Motion Rate

On présente ici une technique de CI itérative permettant d'optimiser certains critères d'optimisation pour la trajectoire articulaire.

La CI itérative permet de trouver numériquement une solution au problème de CI en utilisant les informations contenues dans la jacobienne du manipulateur. Le *Resolved Motion Rate* introduit par Whitney (1969) ainsi que Liegeois (1977) permet d'utiliser la redondance fonctionnelle de la tâche attribuée à un robot initialement non redondant afin de rendre celui-ci redondant et optimiser des critères sur l'espace redondant. On exprime le déplacement incrémental de la façon suivante

$$\Delta q = \underbrace{(J^\dagger T) \Delta x}_{\text{déplacement de la tâche}} + \underbrace{J^\dagger (I - T) J h}_{\text{déplacement redondant}}. \quad (5.4)$$

On utilise le projecteur T permettant de définir l'espace redondant du robot dans l'espace cartésien. Par exemple,

$$T = \begin{pmatrix} I^{3 \times 3} & 0 \\ 0 & 0^{3 \times 3} \end{pmatrix}$$

définit le problème de positionnement comme non redondant et celui de l'orientation comme redondant, ou

$$T = \begin{pmatrix} 0^{3 \times 3} & 0 \\ 0 & I^{3 \times 3} \end{pmatrix}$$

définit le problème de positionnement comme redondant et celui de l'orientation comme non redondant. On peut exprimer différents critères à optimiser sur l'espace redondant. Huo and Baron (2008) ont notamment défini deux critères dans leur article, l'un pour l'évitement des limites articulaires et l'autre pour l'évitement des singularités qui sont décrits en annexe D.

5.3 Alternance des types de cinématique inverse

La CI analytique et la CI itérative sont deux façons distinctes de calculer la CI d'un robot, chacune avec leurs avantages et inconvénients. Pour la CI analytique, la solution se calcule rapidement car elle se trouve analytiquement, mais est vulnérable aux singularités et limites articulaires. Dans le cas de la CI itérative, obtenir la solution de façon itérative en optimisant un critère d'évitement de singularité, de limites articulaires, ou autre. Ceci confère un avantage à l'algorithme de CI itérative rapport au modèle analytique. La CI itérative n'est en revanche pas aussi rapide en terme de temps de calcul que la CI analytique.

Nous pouvons tirer parti de la rapidité de la CI analytique tout en permettant un évitement des singularités avec la CI itérative. Nous allons utiliser les solutions des deux méthodes ; nous utiliserons la CI analytique quand le robot est loin de singularités (on désactive la CI itérative dans ce cas). À l'approche de singularité, le système change de solution pour utiliser la CI itérative comme l'évitement des singularités. Cette méthode permet de profiter de la rapidité du calcul de la CI analytique tout en bénéficiant de certaines optimisations avantageuses de la CI itérative. On observe à la figure 5.6 une trajectoire reproduite. Nous prenons comme indicateur d'approche de singularité le nombre de conditions limites de l'Eq. D.4. Nous prendrons pour la trajectoire choisie

$$\omega_{\max} = 50.$$

Au-delà de ce seuil, nous utiliserons la CI itérative afin d'éviter tout problème de singularité. L'évolution de ω_{ps} le long de la trajectoire et l'activation de la CI itérative est illustré en figure 5.7. Les résultats de la CI est illustré en figure 5.8. On observe en figure 5.9 les temps de calcul pour la structure logicielle complète. On observe qu'utiliser un système de CI mixte permet un gain en terme de temps de calcul non négligeable par rapport à la CI itérative.

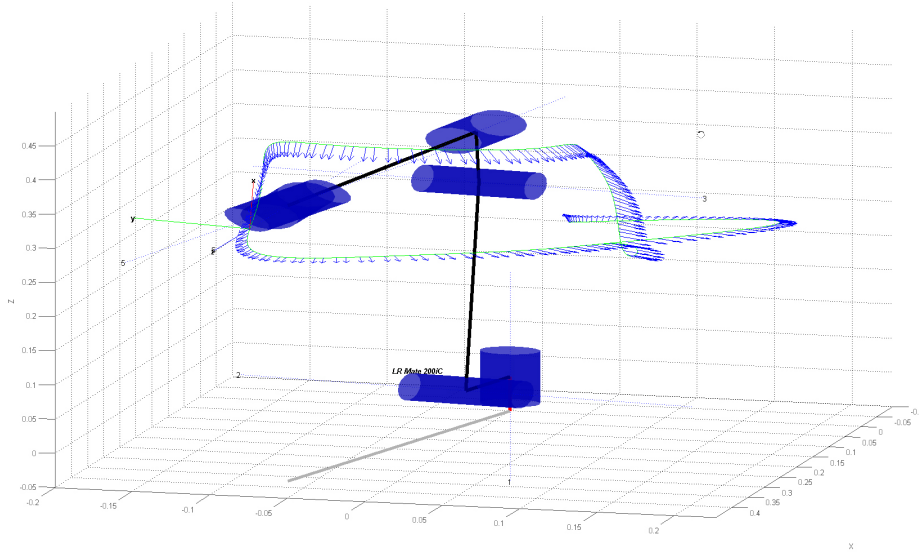


Figure 5.6 Trajectoire reproduite pour le test de CI mixte.

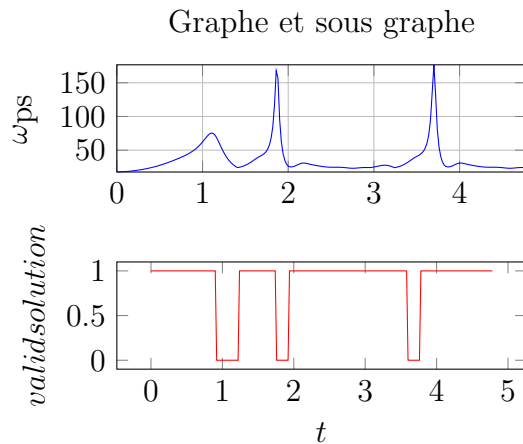


Figure 5.7 ω_{ps} pour une trajectoire d'exemple avec CI mixte. La limite a été fixée ici à $\omega_{max} = 50$. Au-delà de cette valeur, le résultat de la CI itérative est choisie ; sinon, on prend le résultat de la CI analytique.

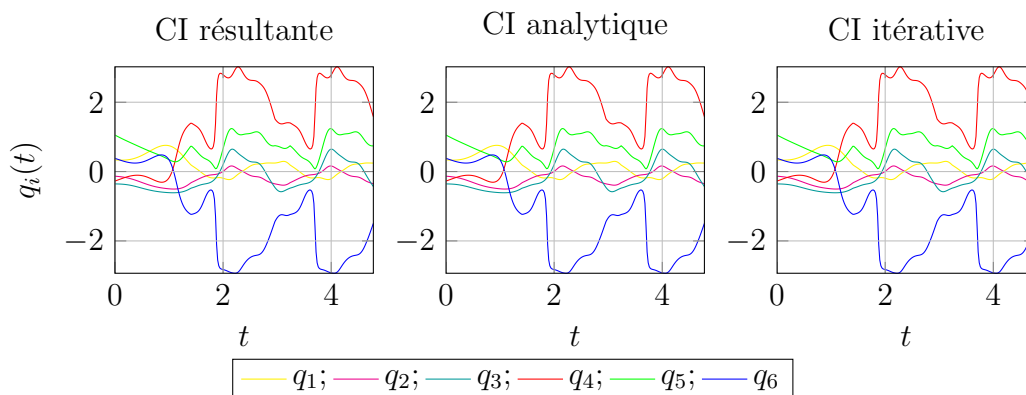


Figure 5.8 Résultat de la CI mixte (gauche), analytique (milieu) et itérative (droite). Les résultats sont identiques.

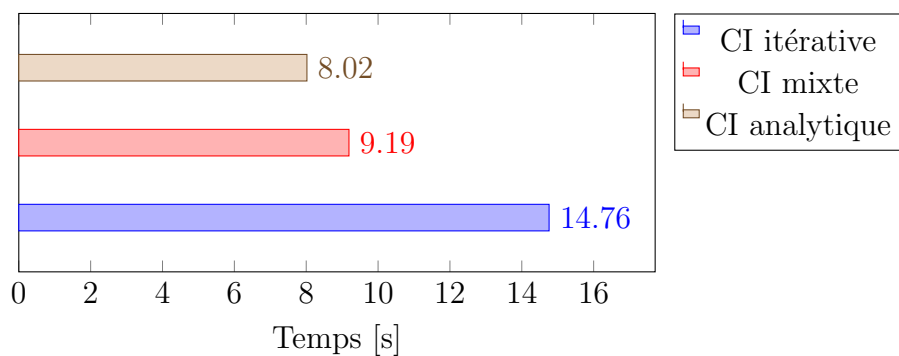


Figure 5.9 Comparaison des temps de calcul entre la CI analytique, itérative et mixte.

5.4 Mélange de MJ cartésien et articulaire

Hersch and Billard (2008), Pattacini et al. (2010) et Pattacini et al. (2010) ont réalisés un contrôleur pour robot manipulateur permettant de réaliser des trajectoires point à point respectant un critère hybride dans le domaine articulaire aussi bien que cartésien. De façon analogue, nous développons ici un système de MJ hybride afin de générer une trajectoire naturelle respectant un critère qui est une pondération entre le MJ dans les domaines articulaire et cartésien (Fig.5.10).

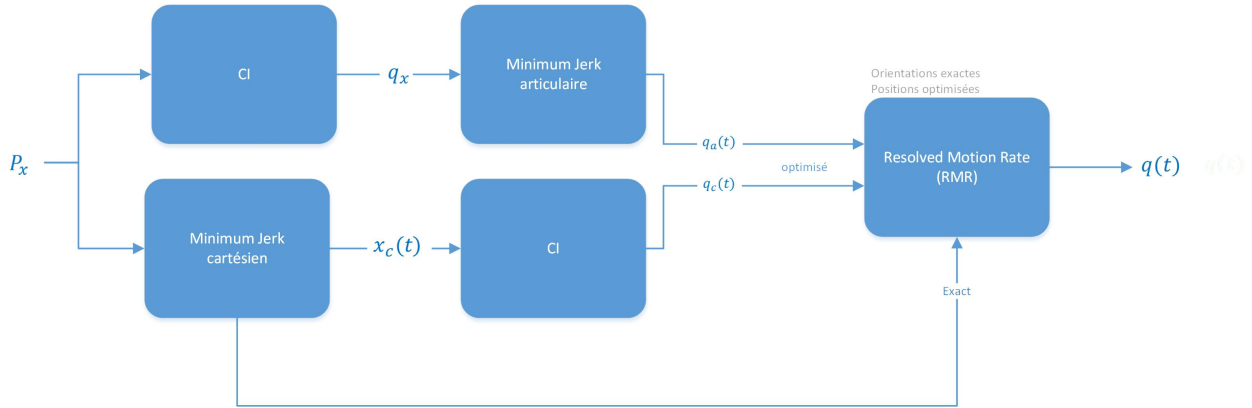


Figure 5.10 Schéma de l'optimisation mixte du MJ cartésien et du MJ articulaire.

L'implémentation développée ici se distingue de celle des auteurs cités ci-dessus dans le sens où elle permet à la fois d'imposer qu'une partie du problème soit résolue exactement et optimiser les DDL redondants selon un critère hybride qui est une pondération de trajectoire de MJ articulaire et/ou cartésien. Ceci permet de réaliser une trajectoire dans le domaine cartésien tout en essayant de minimiser le jerk articulaire et de rendre le mouvement du bras naturel. Ceci est fait dans l'optique qu'un mouvement naturel est optimisé dans le domaine articulaire (Flash and Hogan, 1985), bien que les mouvements complexes sont habituellement leur signification dans le domaine cartésien (e.g. écriture de lettres). De la même façon que Huo and Baron (2008) ont développés les critères d'optimisation d'évitement de limite articulaire et d'évitement de singularité, nous allons ici développer un critère de mélange de trajectoire de jerk dans les domaines articulaires et cartésien.

Soit J_c^* la valeur optimale de la fonction de coût pour une trajectoire optimisée selon le MJ en cartésien, et J_a^* la valeur optimale de la fonction de coût d'une trajectoire optimisée selon

le MJ articulaire. On a les inégalités suivantes

$$\begin{aligned} J_c[q_c(t)] &= J_c^* \leq J_c[q_a(t)] \\ J_a[q_a(t)] &= J_a^* \leq J_a[q_c(t)], \end{aligned} \tag{5.5}$$

avec $q_c(t)$ la trajectoire articulaire minimisant le jerk dans le domaine cartésien, et $q_a(t)$ la trajectoire articulaire minimisant le jerk dans le domaine articulaire. On exprime aussi les positions dans le domaine cartésien par la relation

$$\begin{aligned} x_c(t) &= F[q_c(t)] \\ x_a(t) &= F[q_a(t)], \end{aligned}$$

$F[.]$ étant les équations de la cinématique directe. Le critère à minimiser permettant de suivre une trajectoire cartésienne peut être défini comme

$$z_c = \frac{1}{2}(q - q_c)^T K_c^T K_c (q - q_c) \rightarrow \min_q \tag{5.6}$$

avec

$$K_c = \text{Diag}(k_c)$$

où k_c est un vecteur de coefficients pondérant le critère d'optimisation. On notera que puisqu'on utilise un critère z dans le domaine articulaire et que le projecteur T est défini dans le domaine cartésien, on utilise la jacobienne J pour passer d'un domaine à l'autre, un tel critère d'optimisation ne donnera pas un minimum à $q = q_c$ comme cela serait le cas si $JTJ^{-1} = I$ (c-a-d $T = I$).

On a le gradient de 5.6

$$h_c = -\nabla z_c = K_c(q_c - q).$$

De la même façon, le critère à minimiser permettant de suivre une trajectoire articulaire peut être défini comme

$$z_{joint} = \frac{1}{2}(q - q_a)^T K_a^T K_a (q - q_a) \rightarrow \min_q.$$

Et on a

$$h_a = -\nabla z_a = K_a(q_a - q).$$

En rassemblant les deux critères ensemble, on obtient le critère hybride

$$z_{MJM} = \alpha z_c + (1 - \alpha) z_a \quad (5.7)$$

$$= \alpha \left(\frac{1}{2} (q - q_c)^T K_c^T K_c (q - q_c) \right) + (1 - \alpha) \left(\frac{1}{2} (q - q_a)^T K_a^T K_a (q - q_a) \right) \rightarrow \min_q \quad (5.8)$$

$$h_{MJM} = -\nabla z_{MJM} = -\frac{\partial z_{MJM}}{\partial q} = \alpha K_c (q_c - q) + (1 - \alpha) K_a (q_a - q), \quad (5.9)$$

avec J_{MJM} le critère d'optimisation pour le problème de Minimum Jerk Mixte. Nous prenons ici

$$K_c = K_a.$$

De cette façon, la pondération entre les deux problèmes d'optimisation est entièrement déterminé en imposant la valeur de α . Une valeur de α proche de 1 optimisera surtout le jerk dans le domaine cartésien, tandis qu'une valeur de α proche de 0 optimisera surtout le jerk dans le domaine articulaire.

Regardons l'impact sur la naturalité de la trajectoire dans le cas où nous choisissons arbitrairement de résoudre le problème de positionnement exactement et d'optimiser l'espace nul de ce problème, c-a-d. que nous utilisons le projecteur

$$T = \begin{pmatrix} I^{3 \times 3} 3 \times 3 & 0 \\ 0 & 0^{3 \times 3} \end{pmatrix},$$

et en optimisant au maximum le jerk articulaire ($\alpha = 0$). La trajectoire $q(t)$ dépend du résultat de l'optimisation de l'Eq. 5.4. En suivant exactement la trajectoire de MJ cartésienne pour le problème de positionnement, le jerk de la trajectoire cartésienne reste inchangé et

$$J_c[q(t), \alpha = 0] = J_c^*, \quad (5.10)$$

de par le fait que le MJ en cartésien se calcule uniquement avec le jerk des déplacements en positions (Flash and Hogan, 1985). Ceci est une propriété très intéressante qui nous permet ici de garder la même naturalité de trajectoire selon le critère du MJ dans le domaine cartésien. Dans le domaine articulaire, on a maintenant (en utilisant l'inégalité de l'Eq. 5.5)

$$J_a^* \leq J_a[q(t)] \leq J_a[q_c(t)], \quad (5.11)$$

ce qui permet d'obtenir une trajectoire plus naturelle dans le domaine articulaire tout en gardant la naturalité de la trajectoire dans le domaine cartésien.

Cet exemple est illustré en figure 5.11 pour une trajectoire à $n = 4$ points de passage.

On observe que les positions cartésiennes pour la trajectoire de MJ hybride obtenues sont identiques à celles du MJ cartésien. Les trajectoires articulaires sont cependant différentes. Le jerk cartésien est, comme attendu, identique à $J_c[q_c(t)]$. Le jerk dans le domaine articulaire est quant à lui différent et inférieur à celui de la trajectoire du MJ cartésien.

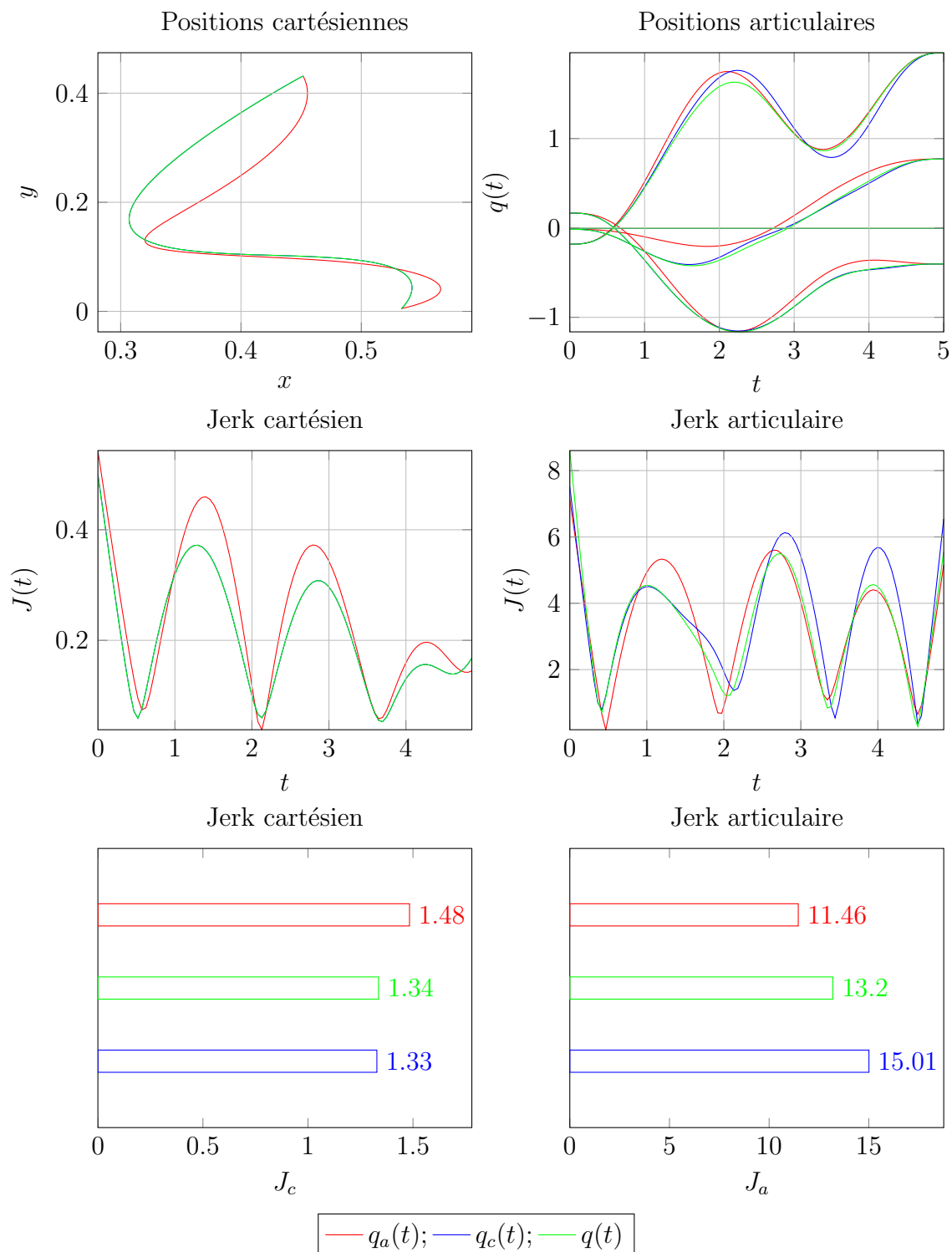


Figure 5.11 Exemple de trajectoire à MJ hybride pour $q_c(t)$, $q_a(t)$ et $q(t)$, résultat de l'optimisation mixte. Analyse dans les domaine cartésien (gauche) et articulaire (droite).

5.5 Conclusion

Nous avons présenté la CI analytique de deux robots industriels ainsi que le choix de la meilleure solution parmi celles possibles. La CI itérative a été présentée avec la technique du RMR, ainsi que deux critères d'optimisation pour le RMR. Une utilisation mélangée à la fois de la CI analytique et de la CI itérative a été présentée afin de tirer parti de la capacité d'optimisation de trajectoires du RMR tout en minimisant les temps de calcul. On observe en pratique une baisse non négligeable des temps de calcul.

Une pondération hybride du critère de MJ entre les domaines cartésien et articulaire a été présentée. Elle permet d'imposer exactement le suivi de la trajectoire de MJ dans les domaines cartésien ou articulaire, et d'optimiser de façon hybride la trajectoire avec les DDL redondants. Ceci permet par exemple de suivre exactement une trajectoire de MJ en positions cartésiennes tout en optimisant les DDL restants (orientation) de la tâche afin d'optimiser le jerk dans le domaine articulaire, ce qui permet de garder la naturalité dans le domaine cartésien en terme de MJ tout en améliorant la naturalité des mouvements dans le domaine articulaire.

CHAPITRE 6

Mise en oeuvre sur un contrôleur industriel

Ce chapitre décrit l'adaptation du contrôle d'un robot manipulateur industriel pour la reproduction d'une trajectoire naturelle. L'adaptation pour un robot FANUC et ABB est expliquée, et nous discuterons de la qualité de la reproduction par rapport à des modèles de contrôles adaptés pour la reproduction de trajectoires naturelles.

La commande d'un robot manipulateur se fait à la base au niveau des couples articulaires

$$u = \tau(q, \dot{q}, \ddot{q}, q_r, \dot{q}_r, \ddot{q}_r)$$

avec u la commande envoyée au robot, q et ses dérivées, le vecteur des positions, vitesses et accélérations articulaires respectivement et q_r et ses dérivées, le vecteur des positions, vitesses et accélérations articulaires de référence, respectivement. Flash (1987) ont décrit qu'une trajectoire naturelle planifiée est reproduite avec un comportement d'un système de masse-amortisseur. Un contrôle proportionnel-dérivé avec compensation de gravité ou un contrôle par couple précalculé sont deux contrôleurs typiques en robotique qui possèdent chacun une composante proportionnelle et dérivée adéquate pour ce genre de contrôle.

En pratique pour un robot industriel, il n'est pas possible d'accéder directement aux couches de contrôle de façon suffisante pour réaliser un tel contrôle. Il n'est de plus pas possible de fournir une trajectoire continue à suivre, ni le profil complet (vitesses, accélérations) de la trajectoire désirée. Le contrôle se fait en fournissant une série de points atteints en spécifiant la précision afin d'atteindre chaque point. Ceci est valable pour la plupart des robots industriels actuels non modifiés (FANUC, ABB, KUKA(Keiser, 2013), Stäubli(Audouin and Stegemann, 2013)). Cette section présente les solutions développées pour les robots FANUC et ABB.

Pour chacun des deux cas, il s'agit de spécifier des points qui seront en temps normal atteints avec un profil de vitesse trapézoïdal. Il est cependant possible de spécifier de ne pas atteindre un point exactement mais de façon approximée (Fig. 6.1).

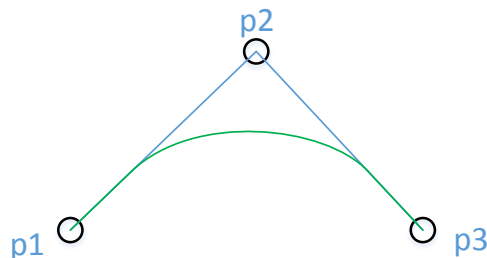


Figure 6.1 Principe du contrôle avec un robot industriel.

6.0.1 Robot FANUC

Avec le contrôleur R-30iA du FANUC LR Mate 200iC, la façon la plus proche de reproduire un mouvement naturel consiste à spécifier un ensemble de points à atteindre. Ces points peuvent être atteints plus ou moins précisément selon le facteur de proximité CNTxx (Fig. 6.2). Obtenir une trajectoire la plus souple possible et la plus représentative de la trajectoire planifiée consiste à jouer sur le paramètre CNT. Il est à noter que le paramètre CNT n'a pas une signification particulière ; il s'agit d'un facteur quantitatif et non pas qualitatif sur la qualité de l'approche des points spécifiés.

- Une valeur de CNT nulle revient à atteindre un point exactement
- Pour une valeur de CNT élevée donnera une trajectoire très souple qui passera loin des points originellement fournis et sera lissée. La valeur exacte de l'éloignement n'est pas connaissable à l'avance, elle peut seulement être évaluée qualitativement.

On prend une trajectoire simple et on la reproduit pour différentes valeurs de CNT en figure 6.3. On observe que pour CNT0, le robot s'immobilise à chaque point de passage. Pour CNT10 et CNT100, le robot ne s'arrête pas. On observe aussi qu'uniquement la trajectoire pour CNT0 respecte réellement les temps imposés.

On reproduit une trajectoire complexe avec $CNT = 80$ en figure 6.4 et avec $CNT = 100$ en figure 6.5. On observe que la trajectoire avec $CNT = 100$ est loin de la trajectoire originale. Prendre $CNT = 80$ donne un meilleur résultat, cependant quand on regarde le temps pris pour atteindre ces points, on se rend compte que la trajectoire prend environ deux fois plus de temps qu'originellement. On choisira dans ce cas précis d'utiliser $CNT = 100$ qui, bien que ne reproduisant pas le profil de positions fidèlement, donne des temps de passage plus cohérents.

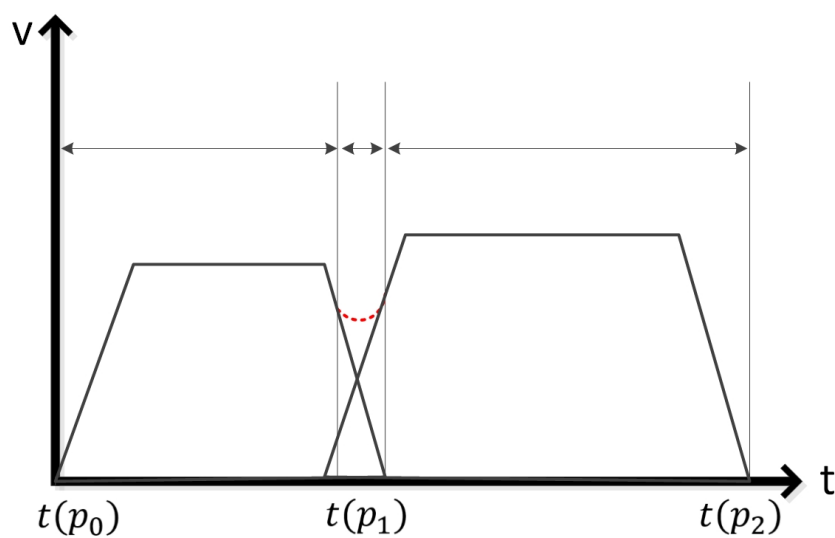


Figure 6.2 Profil de vitesse avec CNTxx pour un contrôleur R-30iA de *FANUC*.

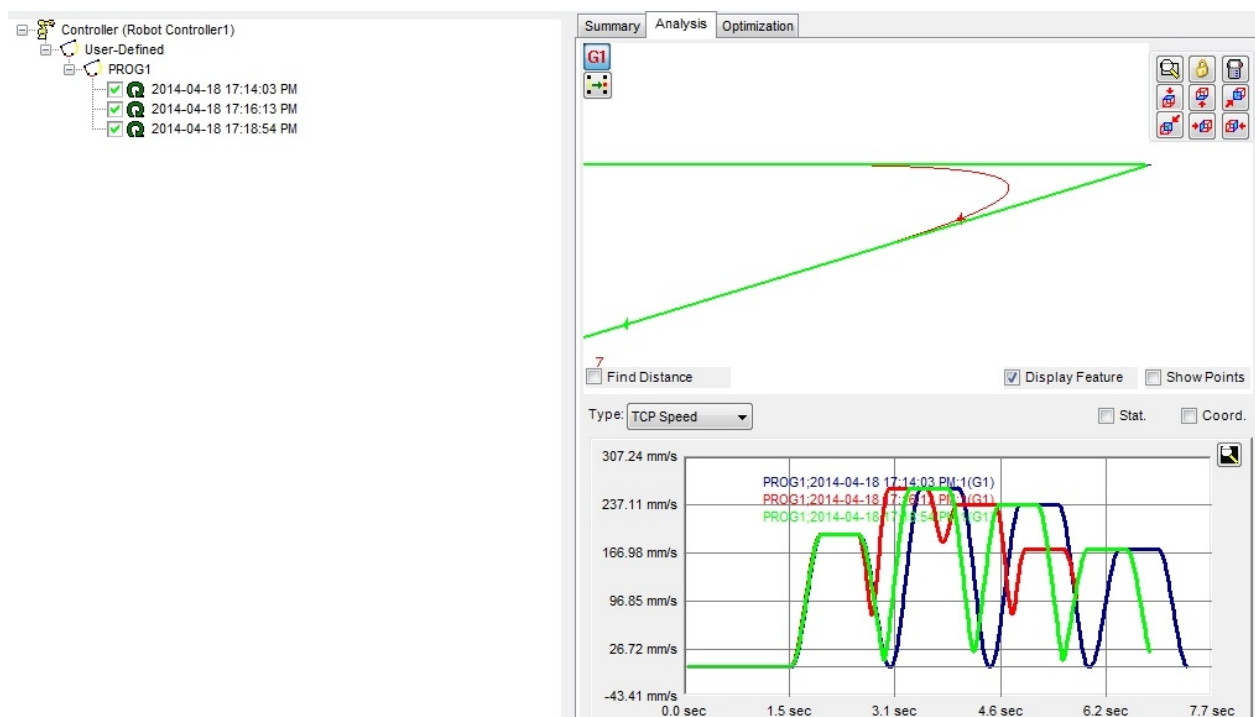


Figure 6.3 Profil de vitesse avec CNT100, CNT0 et CNT10 pour le contrôleur R-30iA de *FANUC*. Résultats obtenu avec le simulateur Roboguide.

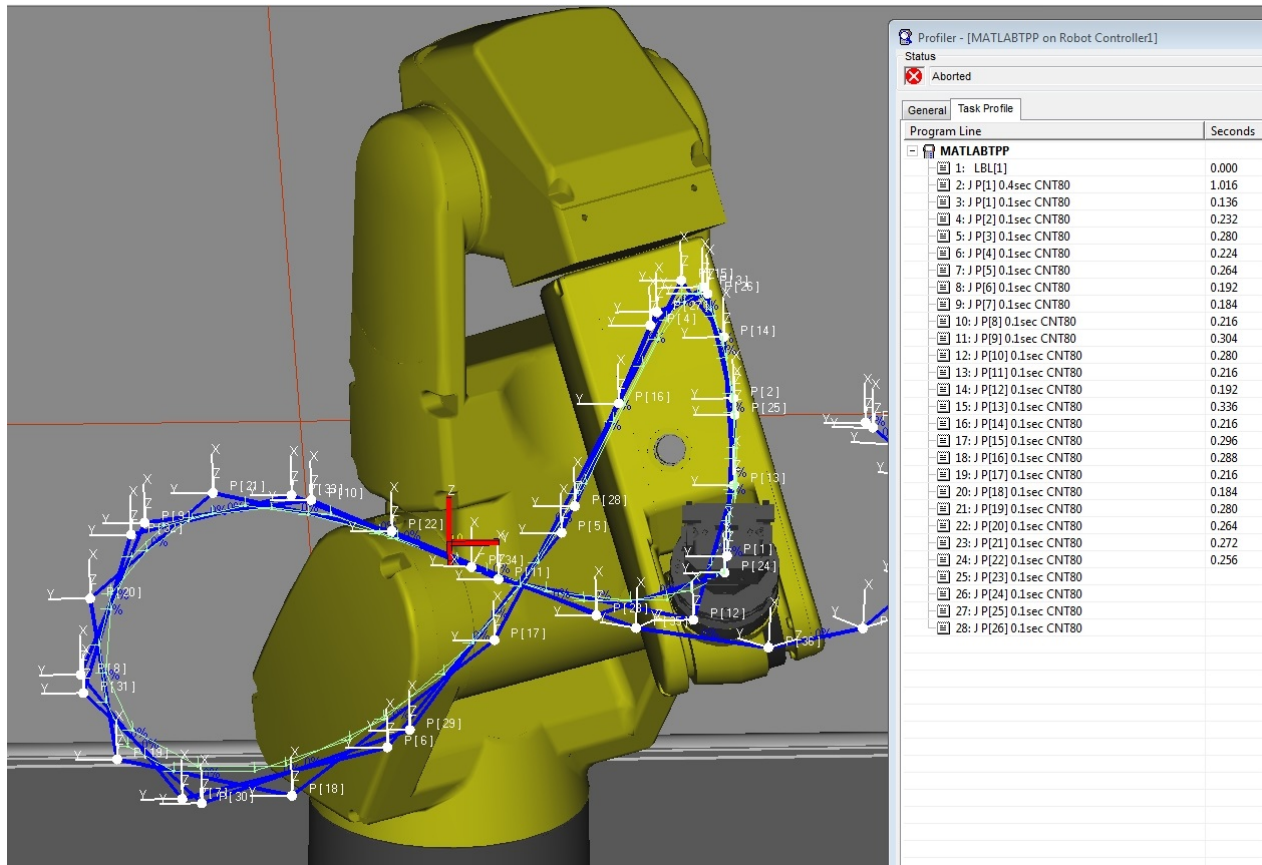


Figure 6.4 Reproduction d'une trajectoire avec paramètre $CNT = 80$. Trajectoire planifiée en bleu, reproduite en jaune. Chaque point doit être atteint en un temps $\Delta t = 0.1$; il est atteint en pratique en un temps $\Delta t \approx 0.2$.

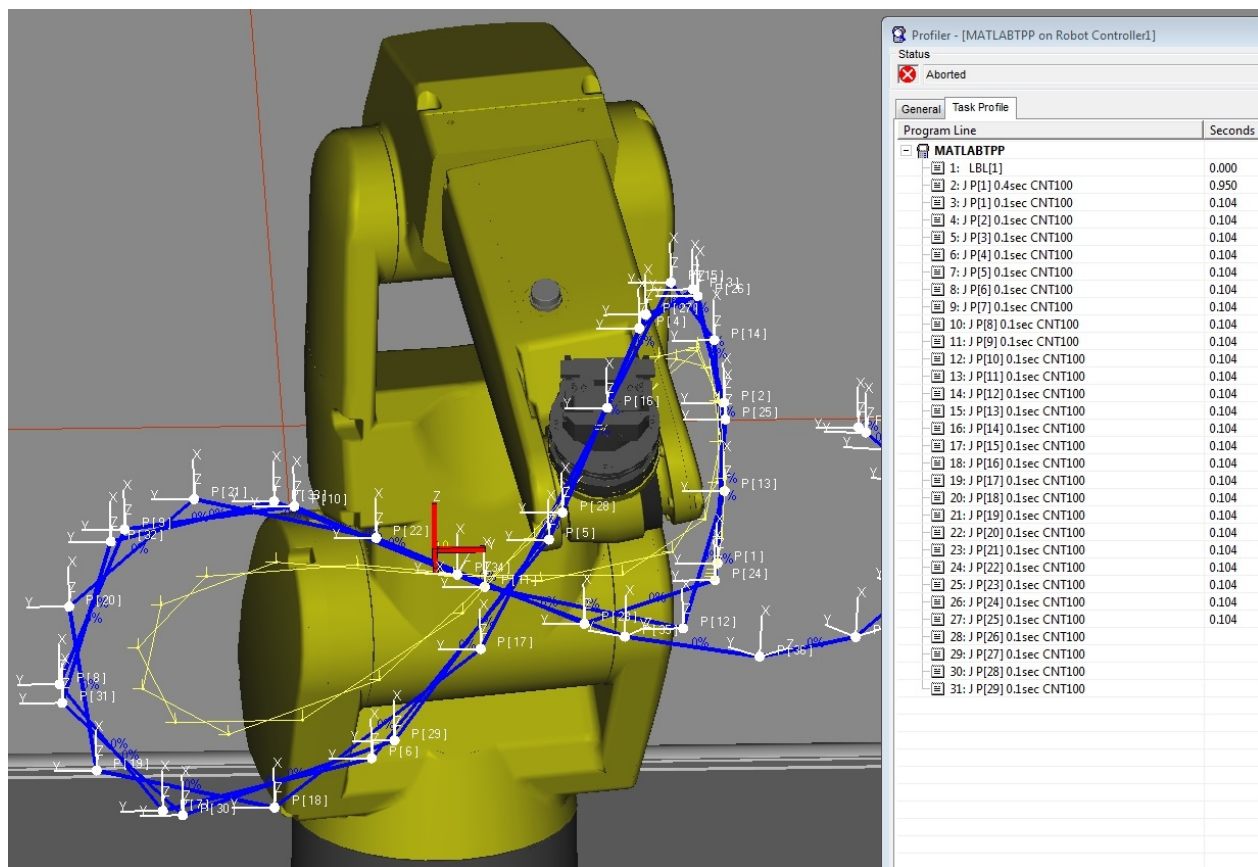


Figure 6.5 Reproduction d'une trajectoire avec paramètre $CNT = 100$. Trajectoire planifiée en bleu, reproduite en jaune. Chaque point doit être atteint en un temps $\Delta t = 0.1$; il est atteint en pratique en un temps $\Delta t = 0.104$.

6.0.2 Robot ABB

Le robot ABB IRB120T possède le même type de contrôle que celui du robot FANUC. À la différence du contrôleur de FANUC, le contrôleur IRC5 de ABB permet de spécifier exactement la distance à partir de laquelle le robot déviara de sa trajectoire exacte. le paramètre permettant de spécifier cela est appelé *zonedata* (Fig. 6.6 et Fig. 6.7). Ces zones permettent de spécifier la distance exacte en millimètres par rapport au point spécifié à partir de laquelle on va dévier de la trajectoire originale afin et se diriger vers le point suivant. Le principe est le même que celui du FANUC et ne sera pas décrit ici.

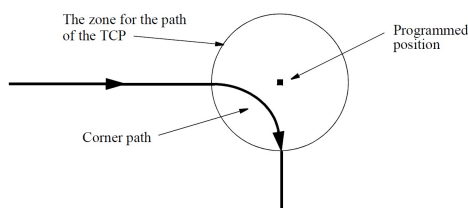


Figure 6.6 Principe du *zonedata* (image issue de Robotics (2010)).

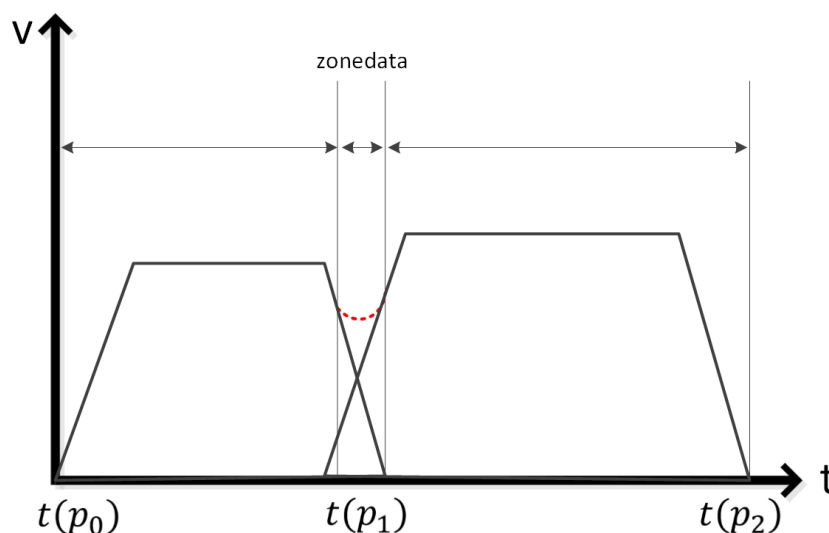


Figure 6.7 Profil de vitesse avec Zonedata pour le contrôleur IRC5 de ABB.

Le contrôleur IRC5 de ABB possède également une fonction supplémentaire, *softservo*, permettant de diminuer l'intensité du contrôle de la position du robot, le rendant plus mou (*soft*). La fonction permet d'adoucir le contrôle; les points à atteindre sont atteints moins précisément, mais les mouvements sont en contrepartie plus souples. Le contrôle reste exactement le même que décrit précédemment.

CHAPITRE 7

Applications

Ce chapitre décrit trois applications développées en utilisant les techniques décrites dans les précédents chapitres. La section 7.1 décrit un système de reproduction rapide de lettres manuscrites, permettant d’observer la naturalité des transitions générées entre les primitives de mouvement. La section 7.2 permet de reproduire des mouvements obtenus par capture de mouvements avec le système de composition séquentielle ; là encore, les transitions sont observées entre les primitives. On observe ici la qualité des transitions pour des mouvements répétitifs. Enfin, la section 7.3 illustre une chorégraphie réalisée avec un robot manipulateur synchronisé avec une suite logicielle de lecture de médias interactifs, qui dans notre cas utilise une musique.

7.1 Écriture de lettres manuscrites

Une des premières applications réalisées est l’écriture de lettres manuscrites. L’alphabet utilisé a été créé en décalquant des points de passages à partir d’une image d’alphabet. Les trajectoires de MJ résultantes ont été générées afin de former la librairie de lettres dont une partie est illustrée en figure 7.1.

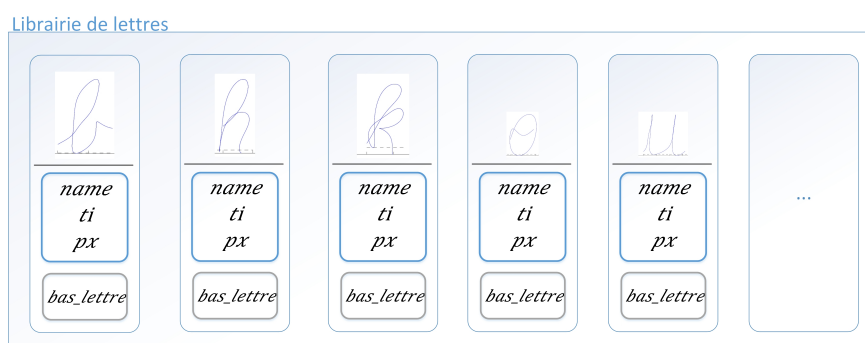


Figure 7.1 Exemple de librairie de lettres générée en MJTO avec leurs caractéristiques stockées. Les éléments entourés en bleu sont les éléments indispensables ; les éléments entourés en gris sont les éléments utilisés par l’utilisateur afin d’imposer le positionnement des primitives lors de la composition séquentielle.

Le but de l’application est d’observer la qualité des transitions générées entre les lettres.

On utilise pour cela le critère d'optimisation pour la durée des transitions comme décrit en section 4.3. Les résultats sont illustrés en figure 7.2 et 7.3 avec le robot FANUC. Pour chacune des expériences, l'écriture en haut est reproduite par le robot, tandis que l'écriture en bas est reproduite par un humain avec une lumière en main.



Figure 7.2 Reproduction du mot "hey" à partir d'une librairie de lettres. L'image a été obtenue par exposition prolongée. Mise en place illustrée en haut.

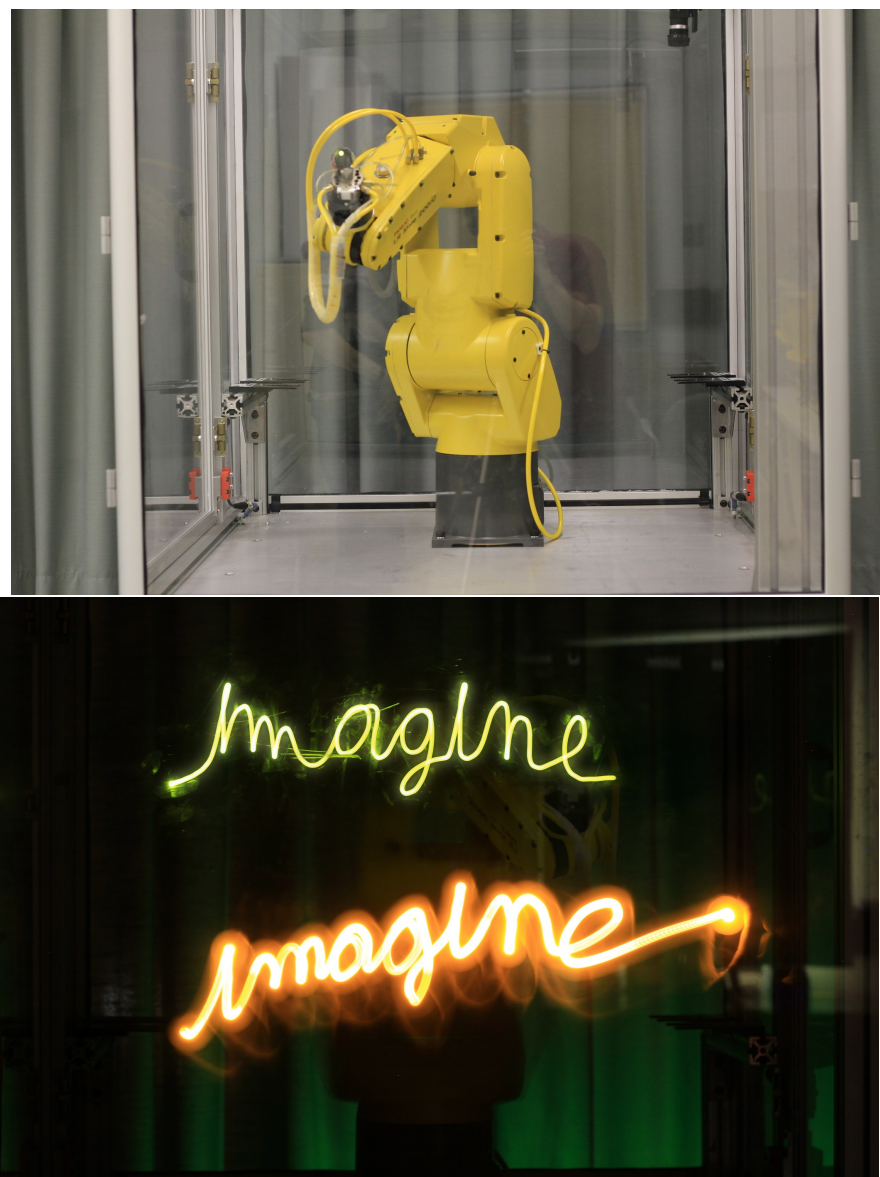


Figure 7.3 Reproduction du mot "imagine" à partir de lettres comme primitives de mouvement. L'image a été obtenue par exposition prolongée. Mise en place illustrée en haut.

7.2 Reproduction de captures de mouvements

Dans la même optique, quelques mouvements de Mocap de base ont été générés avec un système *Vicon* à 6 caméras disponible à l'École Polytechnique de Montréal. Après fitting d'une trajectoire de MJ sur l'enregistrement effectué pour chaque primitive afin de pouvoir récupérer les points de passages ainsi que les t_i nécessaires à la composition séquentielle, on génère la trajectoire dans le domaine cartésien, puis articulaire par la CI. On convertit cela en un fichier .tp reproductible par le robot FANUC. Les résultats sont illustrés en figure 7.5 et 7.6. La mise en place est la même que celle présentée en figure 7.3.

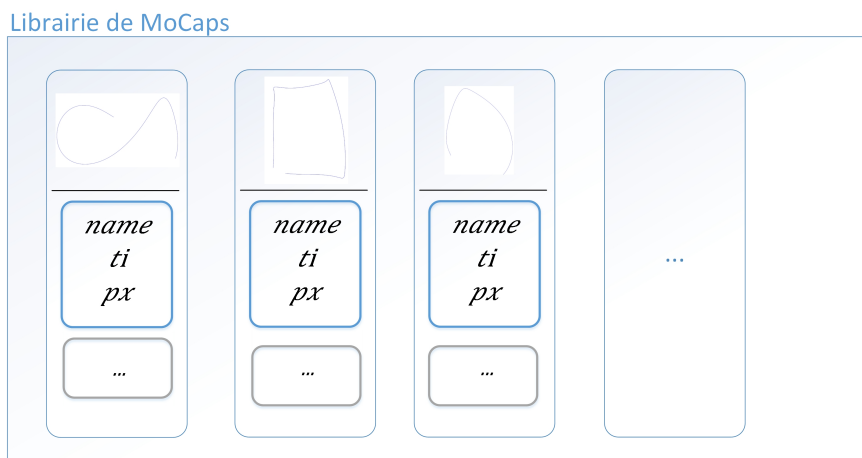


Figure 7.4 Exemple de bibliothèque de mouvements de MoCaps converties avec leurs caractéristiques stockées. Les éléments entourés en bleu sont les éléments indispensables ; les éléments entourés en gris sont les éléments utilisés par l'utilisateur afin d'imposer le positionnement des primitives lors de la composition séquentielle.

Pour chacune des expériences, l'écriture en haut est reproduite par le robot, tandis que l'écriture en bas est reproduite par un humain avec une lumière en main.

7.3 Chorégraphie sur une musique

Une chorégraphie d'un robot synchronisé sur une musique a été réalisée avec la compagnie Moment Factory. Le robot utilisé pour ce projet est le ABB IRB120T décrit en section 5.1.2. Nous utilisons une bibliothèque composée de trois mouvements de base (Fig. 7.7) : un mouvement directement généré à partir du modèle du MJ, et deux autres sont issus d'un mouvement de MoCap placé différemment dans le domaine cartésien pour chacune des deux primitives puis en appliquant la CI.

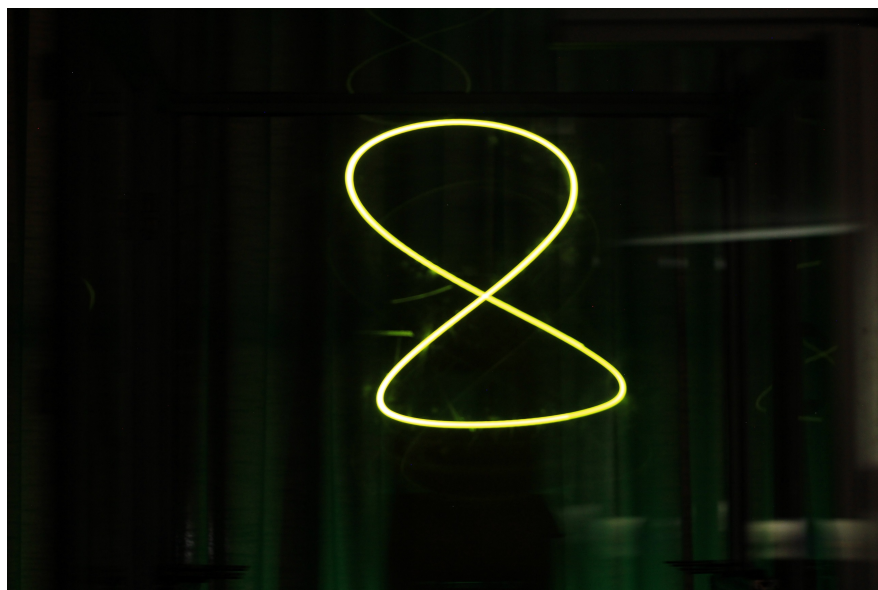


Figure 7.5 Reproduction de mouvements de MoCap. Le début de la MoCap a volontairement été déporté afin de générer deux types de transitions différentes. L'image a été obtenue par exposition prolongée.



Figure 7.6 Reproduction de mouvements de MoCap. Le même mouvement de MoCap que pour la figure 7.5 a été utilisé, mais a été réorienté. L'image a été obtenue par exposition prolongée.

Le but de cette application est double : adapter le robot utilisé avec une suite logicielle appropriée afin qu'il soit commandable de façon interactive et observer les effets du timescaling (décrit en section 4.5) sur les résultats des trajectoires obtenues. L'avantage d'utiliser une librairie de mouvement est de pouvoir s'abstraire le comportement du robot afin de reproduire des trajectoires complexes et d'encapsuler toute la trajectoire en un unique élément.

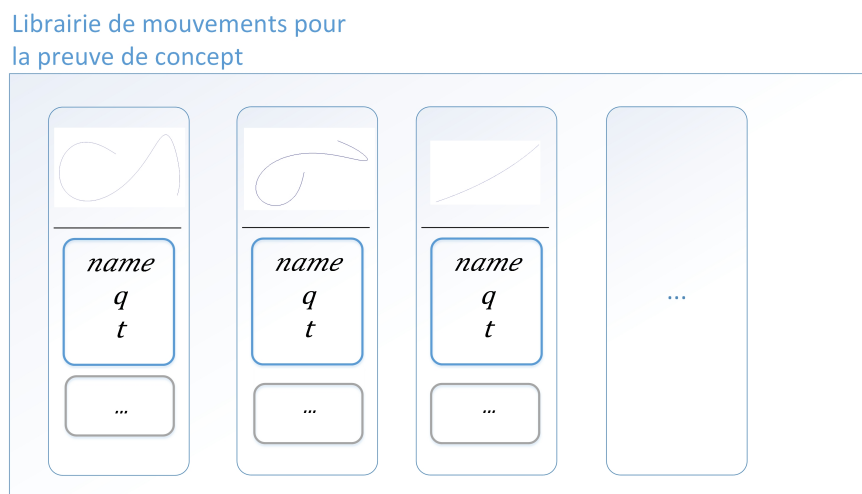


Figure 7.7 Librairie de mouvements pour la preuve de concept.

7.3.1 Implémentation générale

Le système général développé pour ce projet est illustré à la figure 7.8.

7.3.2 Interface avec X-Agora

X-Agora est une suite logicielle pour la gestion et la lecture de médias interactifs créée par Moment Factory. X-Agora comporte notamment un serveur permettant de centraliser l'ensemble des flux d'informations permettant le contrôle de toute installation multimédia. X-Agora comporte également un lecteur permettant de synchroniser plusieurs canaux d'information multimédia (audio, vidéo, autre système). Le lecteur X-Agora permet d'envoyer des commandes au serveur X-Agora afin d'interagir avec n'importe quel appareil multimédia connecté.

Nous utilisons le lecteur X-Agora afin de synchroniser le contrôle du robot avec une musique. Les commandes pour déplacer le robot sont envoyées au serveur X-Agora, qui lui-même envoie ses commandes au robot par protocole TCP.

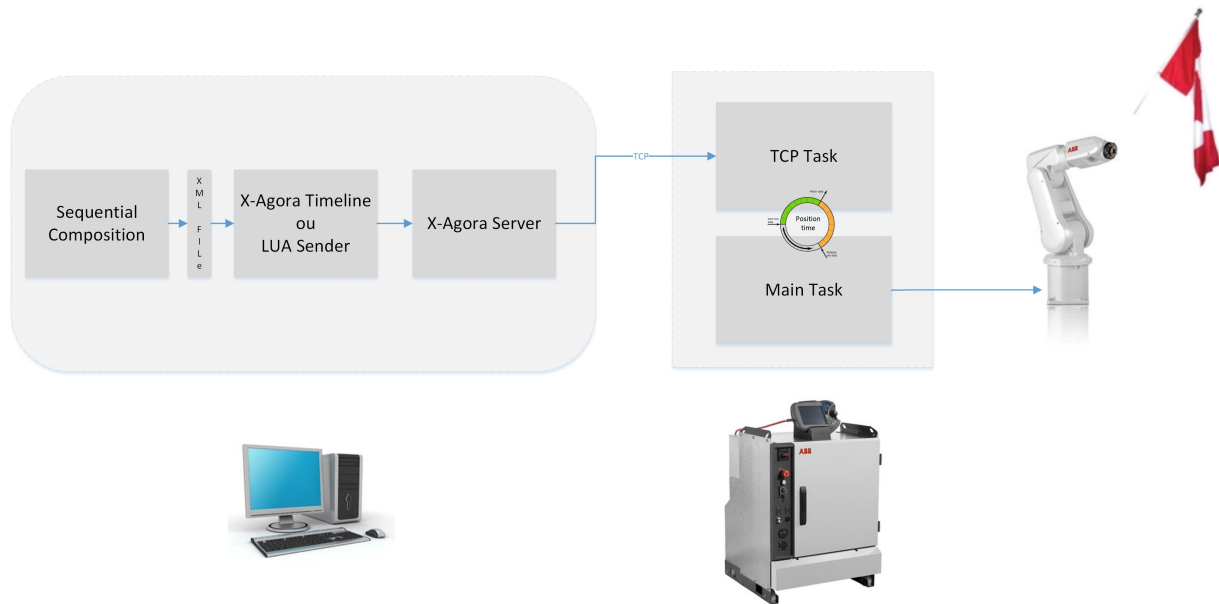


Figure 7.8 Schéma de fonctionnement du système élaboré pour la preuve de concept.

Le système de composition séquentielle génère des trajectoires au format XML. Ces fichiers sont utilisés par le lecteur de X-Agora et envoyée au robot selon les besoins de la chorégraphie.

Deux modes d'envoi de trajectoires ont été implémentés :

- librairie de mouvements dans le robot ;
- émission continue de positions (*streaming*).

La première solution se rapproche le plus du principe original de la composition séquentielle. Une librairie de mouvements est stockée dans le robot, et seule la commande nécessaire pour rejouer chaque primitive est envoyée au robot. Ceci a l'avantage de ne pas nécessiter un gros envoi d'information par TCP, ce qui retire le problème des temps de communication entre le robot et le serveur, qui est généralement un des problèmes principaux dans ce type d'applications (Santiago et al., 2011). L'inconvénient est que les mouvements doivent tous être stockés à l'avance dans le robot, ce qui limite la liberté de création de nouveaux mouvements.

La seconde solution a été testée brièvement. Elle permet effectivement d'envoyer une série de points en continu au robot. Étant donné l'absence de retour d'état afin de pouvoir effectuer un contrôle précis dans le temps, le robot accumule du retard au fur et à mesure de l'exécution de ce système. Ce système n'a pas été testé plus en détail du fait du manque de retour d'état, clairement nécessaire dans ce genre d'opération.

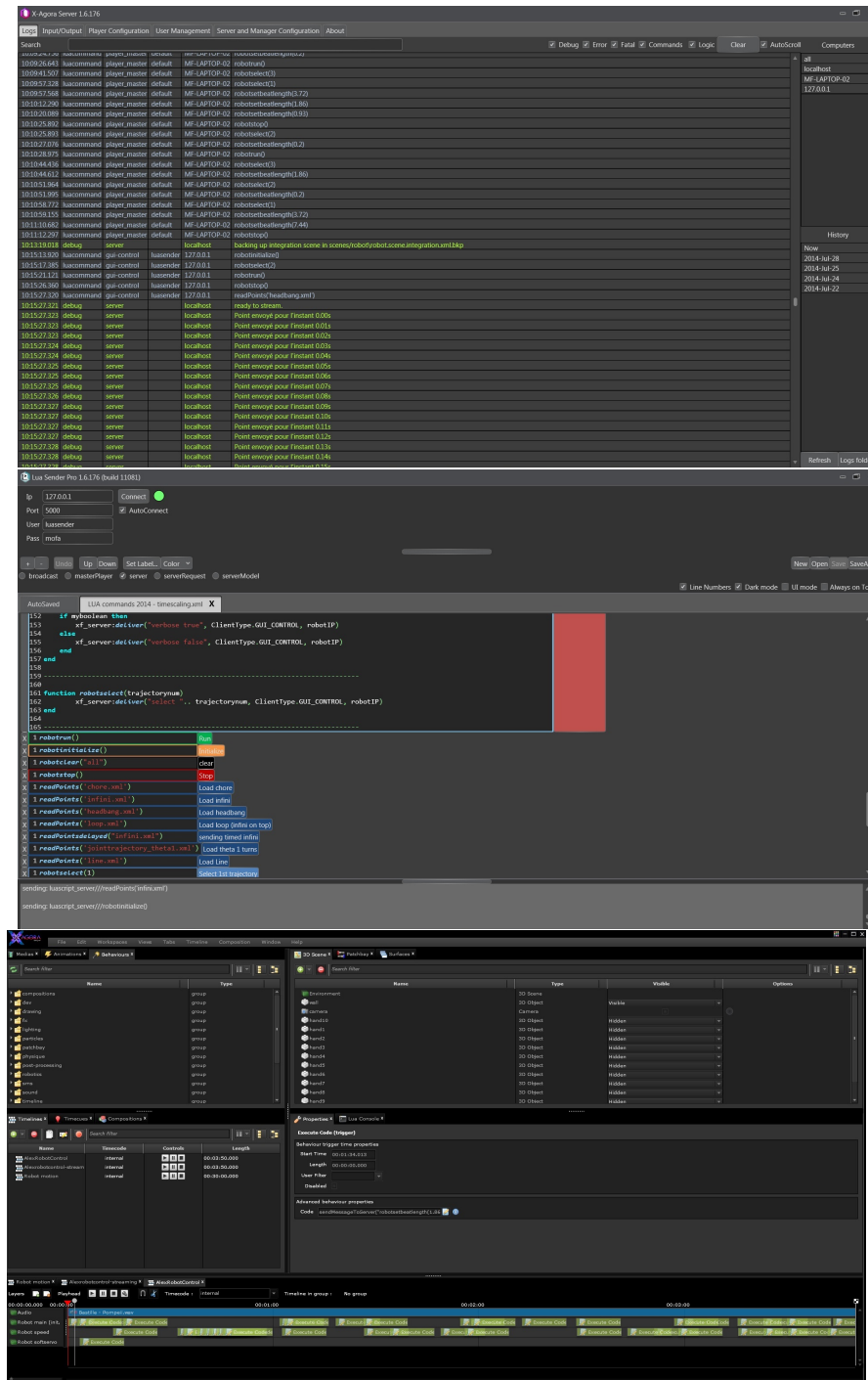


Figure 7.9 Illustration de la suite de logiciels X-Agora. Respectivement : X-Agora Server, X-Agora LUA sender et X-Agora player.

7.3.3 Considérations techniques

Liaison de la structure logicielle de Matlab avec X-Agora

Afin de permettre de récupérer les résultats de la structure logicielle (framework) de la composition séquentielle et de l'utiliser avec X-Agora, il a été décidé d'utiliser des fichiers XML contenant toutes les informations nécessaires à la reproduction des trajectoires désirées. Le format est illustré en figure 7.10. La ligne avec balise *data* permet de récupérer des informations sur l'incrément Δt entre chaque position, ainsi que la durée totale du mouvement t_f . Chaque ligne successive constituant la trajectoire est un point caractérisé par la position articulaire q_0, \dots, q_6 et le temps t donné.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <trajectory>
3    <data dt="0.2" tf="1.20" />
4    <point t="0.00" q1="19.0" q2="23.4" q3="47.6" q4="-82.5" q5="-82.1" q6="22.7" />
5    <point t="0.20" q1="17.8" q2="10.4" q3="48.6" q4="-55.3" q5="-104.6" q6="89.2" />
6    <point t="0.40" q1="9.9" q2="30.0" q3="31.0" q4="-35.2" q5="-76.7" q6="85.5" />
7    <point t="0.60" q1="-1.0" q2="44.0" q3="18.9" q4="-9.5" q5="-59.5" q6="47.7" />
8    <point t="0.80" q1="-6.3" q2="36.6" q3="17.5" q4="7.3" q5="-76.1" q6="6.2" />
9    <point t="1.00" q1="3.8" q2="28.1" q3="24.6" q4="-30.2" q5="-88.2" q6="-18.2" />
10   <point t="1.20" q1="16.1" q2="29.3" q3="39.0" q4="-69.6" q5="-69.0" q6="-9.1" />
11 </trajectory>

```

Figure 7.10 Exemple de fichier XML contenant une trajectoire.

Protocole TCP et envoi de commandes

Le langage utilisé pour envoyer les commandes au serveur X-Agora est le langage LUA aussi bien pour le LUA Sender que pour la Timeline X-Agora. L'envoi d'information au robot se fait par le réseau en utilisant le protocole TCP. Le protocole envoie sur le réseau des paquets dont l'utilisateur n'a pas le contrôle direct de la taille (Fig. 7.11).

Cart	0.000	0.000	1.134	2.368	3.480	0.000	\0	Cart	0.000...
------	-------	-------	-------	-------	-------	-------	----	------	----------

Figure 7.11 Exemple de découpage d'une commande par paquets pour le protocole TCP.

En prenant en compte les spécifications du contrôleur ABB IRC5, une instruction, qui est stockée ici sous la forme d'une variable *string*, est de maximum 80 bytes. Il faut donc s'assurer

- que l'on a reçu l'entièreté de la commande envoyée avant de la traiter ;
- que le message envoyé a une taille inférieure à 80 byte.

Pour le premier cas, en observant que X-Agora rajoute un signe `\0` à la fin de chaque message envoyé, on peut s'assurer de la fin du message en vérifiant l'existence de `\0` dans le message reçu. Dans le cas contraire, on concatène les messages reçus jusqu'à la réception de ce code. Pour le second cas, c'est à l'utilisateur de s'assurer lors de l'envoi que le message est bien inférieur à 80 bytes.

On envoie une commande en X-Agora en utilisant le code de la figure 7.12, avec *command* la commande à envoyer, et *robotIP* l'IP du robot sur le réseau. Au niveau du contrôleur, c'est la tâche secondaire (voir figure 7.8) qui va attendre et recevoir les paquets avec l'instruction du code 7.13. Le paramètre 80 permet de limiter la taille des paquets à recevoir à 80 bytes, mais ne garantit pas de recevoir une instruction complète, terminée par `\0` (noté `\00` en langage RAPID).

```
1  xf_server:deliver(command, ClientType.GUI_CONTROL, robotIP)
```

Figure 7.12 Envoi de commande en Lua au serveur X-Agora.

```
1  SocketReceive socket \Str := str_data \ReadNoOfBytes:=80 \Time := WAIT_MAX;
```

Figure 7.13 Réception de paquets TCP en RAPID.

Implémentation dans le contrôleur

Le robot doit pouvoir effectuer deux tâches en parallèle : déplacer le robot, et écouter le réseau afin de recevoir les commandes à accomplir. Ceci se fait en parallèle grâce au contrôleur IRC5 de ABB. Celui-ci tourne sur une monture de Windows CE qui permet le multitâches. Le multitâches du IRC5 semble ne pas être du vrai parallélisme mais plutôt du pseudo-parallélisme (ou temps partagé) : plusieurs situations ont été observées où le robot ne se déplaçait plus, mais semblait recevoir tous les messages du réseau, ou encore d'autres situations où le robot se déplaçait mais n'écoutait plus les messages du réseau.

Il faut pour empêcher cela imposer un temps de repos (instruction *WaitTime*) à un moment donné pour chaque tâche afin que l'ordonnanceur puisse faire tourner les deux tâches.

Pour le déplacement du robot, on utilise la commande *MoveAbsJ*, qui permet de spécifier la *zonedata* ainsi que le temps requis pour atteindre le point (Fig. 7.14).

```
1 MoveAbsJ jointtargetbuffer{touse,fetchdataindex{touse}} \NoEOffs, vmax \T:=
    scaleddt, z50, toolflag;
```

Figure 7.14 Commande de déplacement utilisée pour le robot avec ses arguments. *MoveAbsJ* est le mode de déplacement, *scaleddt* est le temps en secondes afin d’atteindre le point, *z50* définit la *zonedata*.

Commandes implémentées

Les commandes implémentées pour la preuve de concept sont illustrées dans le tableau 7.1. Les commandes ne possédant pas d’équivalent en LUA sont en fait utilisées à l’intérieur d’autres fonctions LUA.

Délais de réponse

Lors de la reproduction de la trajectoire, divers délais de différentes origines apparaissent. Le système ne possédant pas de retour d’état concernant le respect du synchronisme, il n’est pas possible de réaliser une correction quant au minutage exact d’exécution du système. Voici une liste non exhaustive des causes de l’augmentation des délais de réponse du robot par rapport à la trajectoire planifiée :

- délais de communication (TCP) ;
- réaction et traitement de la tâche secondaire ;
- réaction et traitement de la tâche principale ;
- synchronisme pour atteindre le point donné au robot irréaliste.

Il est à noter que selon Robotics (2009a), le temps de réponse du protocole TCP est de moins de 100ms. Les gros délais observés sur le robot ne sont donc pas dûs exclusivement ou majoritairement à un délai de communication. Ces délais ne sont pas constants mais bien variables, ils dépendent notamment du code présent dans la tâche principale et secondaire, ce qui ne permet pas de compenser le retard par une constante.

Mise à l’échelle des temps

La mise à l’échelle des temps (*timescaling*) a été ajoutée comme une fonctionnalité exploitable dans le lecteur X-Agora. Les trajectoires étant directement reliées à des temps bien précis, on peut modifier ces temps par un facteur constant afin d’accélérer ou ralentir les mouvements, comme expliqué en section 4.5.

Tableau 7.1 Tableau des commandes LUA et RAPID associées.

Commande LUA	Équivalent RAPID	Description	Arguments LUA
robotrun	run	Démarrer le robot	-
robotstop	stop	Arrêter le robot	-
robotclear	clear	Effacer les données de trajectoires enregistrées dans le robot	-
robotinitialize	initialize	Initialiser le robot à sa position initiale	-
robotstopsecondarytask	stopsecondarytask	Arrêter la tâche secondaire	-
robottimescaling	timescaling	Changer la vitesse d'exécution de la trajectoire planifiée	$\text{timescalingfactor} = \left(\frac{t_f'}{t_f}\right)$
-	settf	Définir la durée du mouvement	tf (durée du mouvement)
robotsetbeatlength	beatlength	Définit la durée du mouvement via le facteur de mise à l'échelle du temps	beatlength(entier)
robotsoftservo	softservo	Définir la souplesse des mouvements du robot	softfactor (entier entre 0 et 100)
robotcanpreparefornext	canpreparefornext	Le robot se préparera (ou non) avant de démarrer	myboolean
-	trajectoryinfo	Afficher des informations sur la trajectoire sur le Teach Pendant	-
robotverbose	verbose	Afficher des informations continues sur le TP	myboolean
robotselect	select	Sélectionner une primitive à jouer	trajectorynum
-	jpos1	Téléverser des informations pour la primitive 1	$q_1, q_2, q_3, q_4, q_5, q_6, t$
-	jpos2	Téléverser des informations pour la primitive 2	$q_1, q_2, q_3, q_4, q_5, q_6, t$
-	jpos3	Téléverser des informations pour la primitive 3	$q_1, q_2, q_3, q_4, q_5, q_6, t$
readPoints	-	Lire une trajectoire .xml et envoyer tous les points au contrôleur	filename (nom du fichier avec extension .xml à lire)
readPointsdelayed	-	Lire une trajectoire .xml et envoyer les points à des moments précis	filename (nom du fichier avec extension .xml à lire)

Pour le mode avec librairie de mouvements téléversée dans le robot, le timescaling peut se faire par envoi direct du facteur d'échelle par TCP afin de modifier la vitesse de reproduction de la trajectoire.



Figure 7.15 Modification de la résolution temporelle d'un mouvement.

Si le mouvement du robot est trop ralenti, la résolution de la trajectoire peut se dégrader et les mouvements devenir saccadés (Fig. 7.15). Afin de pallier à ce problème, on s'assure que les mouvements enregistrés en .XML ont une résolution temporelle suffisante. Nous avons pris un écart de temps Δt entre chaque position de $0.001s$. La résolution maximale de reproduction de trajectoire pour le ABB IRB120T étant de $0.01s$ (Robotics, 2010). Ceci pose un autre problème : si on donne des points trop peu espacés les uns des autres (en particulier, des points dont la distance est inférieure à la *zonedata* utilisée entre chaque point (Robotics, 2009a)), l'algorithme d'interpolation de trajectoire du robot ABB générera des trajectoires plus proches des points de passage qu'avec une résolution plus grande, au lieu de faire une interpolation lisse des points fournis. Il en résultera aussi que les temps pour atteindre ces points ne seront plus exactement respectés. Comme solution, nous imposons un écart de temps minimum de $0.01s$ entre chaque point à reproduire par le robot. Un compteur incrémenté dans une boucle nous permet d'ignorer tous les points présents dans un intervalle $[0, 0.01]s$.

7.3.4 Preuve de concept

La preuve de concept réalisée à Moment Factory est une chorégraphie d'un robot agitant un drapeau en coordination avec la musique *Bastille* de *Pompeii*. Toute la chorégraphie est d'abord paramétrée dans la ligne de temps (*timeline*) X-Agora (Fig. 7.16). On envoie au tout début trois primitives de mouvements au robot, qui les enregistre dans le contrôleur IRC5 dans des buffers respectifs. Des commandes permettant de démarrer, arrêter, sélectionner un mouvement ou modifier la vitesse d'exécution sont ensuite envoyées afin de créer la chorégraphie. Une vidéo a été réalisée pour la chorégraphie, dont une illustration est disponible en figure 7.17).

Les temps exacts d'exécution par le robot ne sont pas exactement ceux spécifiés dans la timeline, pour les raisons expliquées précédemment. Les temps dans la ligne du temps ont été ajustés à la main afin de pallier à ces retards. De plus, lors de la chorégraphie, le robot s'arrête

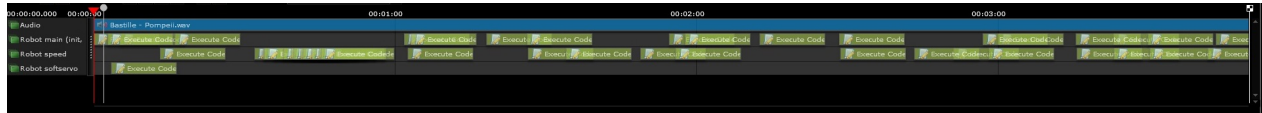


Figure 7.16 Illustration de la timeline X-Agora pour la preuve de concept.



Figure 7.17 Illustration de la mise en place du robot pour la preuve de concept.

volontairement à certains moments prédéfinis, ce qui permet d'éliminer l'accumulation des retards.

7.3.5 Discussion

Il est à noter que pour la dernière application, la structure logicielle de la composition séquentielle n'a pas été appliquée complètement. Les transitions, par exemple, n'ont pas pu être incorporées correctement par manque de temps. Afin de prendre en compte les transitions telles qu'implémentées précédemment, il faudrait générer une librairie de toutes les transitions possibles et les téléverser dans le robot de la même façon que la librairie de primitives.

On peut observer à la dernière application qu'avec une loi de contrôle non-appropriée, on observe que le robot vibre et que ce comportement n'a pas été modélisée précédemment, ce qui détériore l'apparence des mouvements générés. Il serait intéressant de voir comment ce comportement peut s'incorporer au niveau de la génération de trajectoires naturelles afin de prendre celles-ci en compte.

Techniquement parlant, cette preuve de concept pourrait être ultérieurement améliorée afin de permettre une synchronisation avec de l'équipement supplémentaire, par exemple un projecteur afin de réaliser une application interactive. Synchroniser un projecteur pour le cas d'une application de projection mapping peut se faire de façon interactive à partir du système implémenté en récupérant la position réelle du robot de façon périodique, par exemple avec le système *Robot Reference Interface* (RRI) du robot ABB utilisé (Fig. 7.18), et en l'utilisant afin de connaître en tout temps la position des éléments sur lesquels projeter.

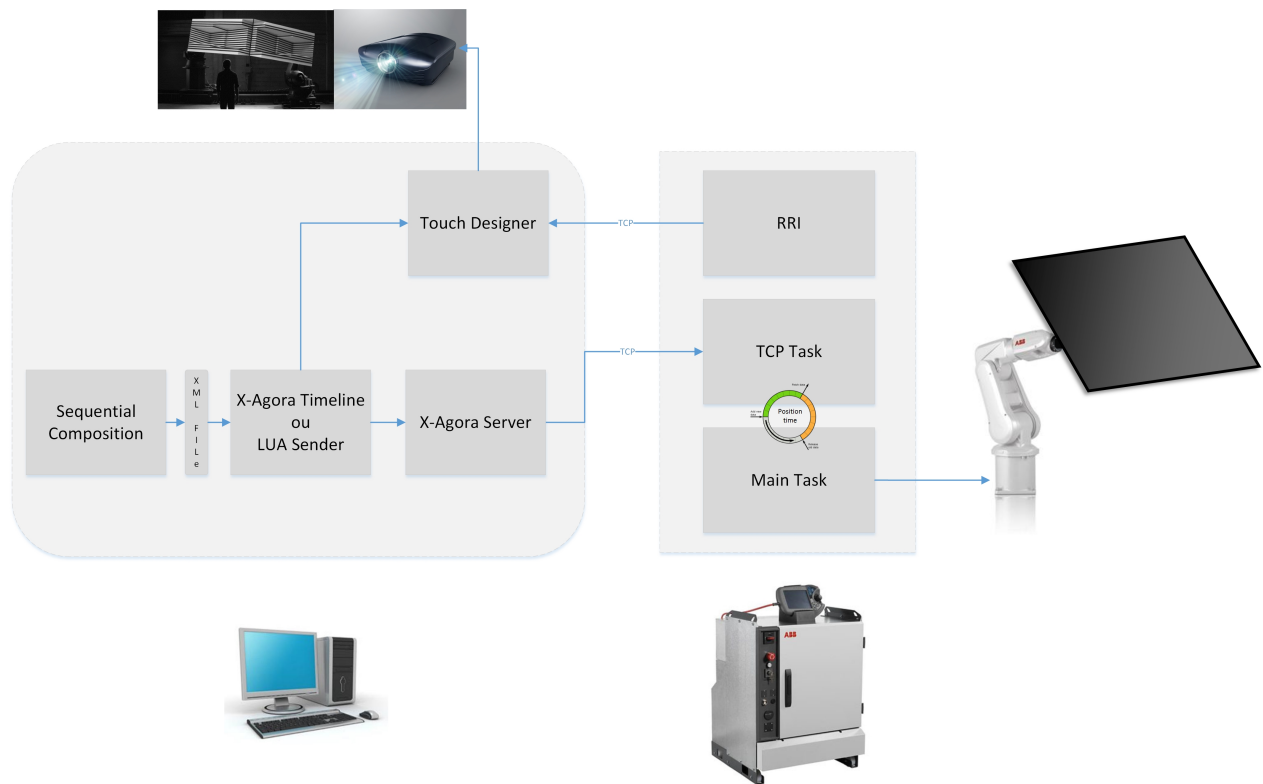


Figure 7.18 Exemple d'extension du robot synchronisé pour une application avec projection mapping.

CHAPITRE 8

CONCLUSION

8.1 Synthèse des travaux

Une structure logicielle a été développée afin de séquencer des mouvements interactivement entre eux et comprend différents outils permettant de séquencer et générer des trajectoires naturelles. Ces techniques répondent à différents besoins, de l'apparence de la trajectoire au besoin de reproduction en temps réel.

L'ensemble de ces techniques est basé sur le séquençage de primitives de mouvement. On assemble des points de passages de problèmes de MJ de plusieurs primitives de mouvement afin de créer un nouveau problème de MJTF pour une trajectoire plus complexe. Le système développé peut être solvable en temps réel si les temps de transitions sont imposés par l'utilisateur. La naturalité de la trajectoire globale est conservée.

La trajectoire ainsi générée donne une trajectoire légèrement différente des trajectoires des primitives originales ; l'optimisation se faisant sur toute la trajectoire, les vitesses et les accélérations aux points de passages sont optimisées pour chaque assemblage. Cela correspond implicitement à des trajectoires à caractéristiques humaines où le même mouvement n'est jamais réalisé exactement de la même façon. Cette optimisation se produit directement au niveau de la planification de la trajectoire.

8.2 Limitations de la solution proposée et améliorations futures

Le système de contrôle utilisé pour les applications sur des robots industriels n'est pas un contrôle idéal pour une trajectoire naturelle ; la consigne est un ensemble de points à atteindre et ne permet pas d'utiliser des méthodes de contrôle adaptées pour la reproduction de trajectoires naturelles. Les contrôleurs proportionnel-dérivé avec compensation de gravité ainsi que le contrôleur à couple précalculé ont été cités. Ces deux contrôleurs requièrent une commande en couple. Un autre mode de contrôle possible serait le contrôle avec commande en vitesse. Les robots ABB permettent pour le moment de contrôler un unique axe en vitesse (Robotics, 2010). Un tel contrôle a déjà été testé pour des robots KUKA pour des consignes de vitesses constantes par Keiser (2013). Il serait intéressant de porter le concept pour des consignes de

vitesse continues pour chaque axe du robot et d'observer la qualité de la reproduction d'une trajectoire naturelle.

Les temps de calcul de la CI n'ont pas été pris en compte dans les temps de calcul montrés pour la composition séquentielle (chapitre 4) ; seuls les temps de génération de la trajectoire ont été pris en compte. La cinématique inverse peut néanmoins se faire en temps réel, et des techniques ont été présentées permettant d'améliorer la rapidité des algorithmes classiques.

La structure logicielle implémentée ne permet pas immédiatement d'effectuer des applications en temps réel dans ce sens qu'elle demande au préalable l'ensemble des primitives à calculer, lesquelles sont ensuite séquencées en temps réel afin de créer un comportement réellement interactif. Il faudrait créer un système multi processus où les mouvements à reproduire seraient fournis continuellement au système. Un tel système a été jugé trop long à mettre en place dans le cadre de cette recherche mais pourrait pleinement tirer parti de la structure logicielle développée ici pour des applications interactives.

Les points de passage pour le problème de MJ sont des contraintes arbitraires imposées par l'utilisateur. Comme expliqué au chapitre 2, de telles contraintes ne sont pas encapsulées par le critère d'optimisation pour former une trajectoire naturelle. Incorporer ces contraintes comme un problème à optimiser dans le critère optimisé permettrait de s'affranchir de ces contraintes dures. Le critère du MJ passerait donc de celui de l'Eq. 3.1 à

$$J(x(t)) = \alpha \frac{1}{2} \int_{t_0}^{t_f} \left| \frac{d^3 x(t)}{dt^3} \right|^2 + \beta \frac{1}{2} \sum_{i=1}^n |x(t_i) - x_i|^2 dt$$

avec n le nombre de points de passages et x_i les points de passage au temps t_i . Les coefficients α et β servent à pondérer les deux critères.

On a également supposé que le robot est formé de corps rigides pour chacune de ses articulations ; on observe sur les applications du chapitre 7 qu'avec une loi de contrôle non-appropriée, on peut observer que le robot vibre d'une façon non-modélisée précédemment, et qui dégrade l'apparence des mouvements générés. Il serait intéressant de voir comment ce comportement peut se modéliser au niveau de la génération de trajectoires naturelles afin de prendre celles-ci en compte.

8.3 Conclusion

Pour conclure, ce projet fortement multidisciplinaire a permis d'explorer différents aspects du séquençage interactif de trajectoires naturelles pour un robot manipulateur. Le séquençage de ces mouvements se base sur l'approche par primitive de mouvements qui est de plus en plus utilisée afin de pouvoir reproduire efficacement des comportements complexes. L'accent est mis ici sur l'expression de trajectoires en tant que problème d'optimisation, et les méthodes possibles afin de résoudre ces problèmes le plus rapidement possible. La reproduction de trajectoires naturelles de façon interactive prend ses sources dans tous les domaines qui touchent aux robots manipulateurs, depuis la robotique, le contrôle et l'optimisation, jusqu'à la structure logicielle régissant son utilisation. Tous les domaines y sont interdépendants et leur synergie est seule capable de donner naissance aux projets les plus ambitieux. Que la robotique soit considérée simplement pour une utilisation industrielle, artistique voire biomécanique, elle reste un centre d'intérêt passionnant et plein d'évolution. Comme le disait *Isaac Asimov*(1920-1992), l'auteur des trois lois de la robotique :

« L'être humain croira toujours que plus le robot paraît humain, plus il est avancé, complexe et intelligent »



RÉFÉRENCES

- M. Al Borno, M. de Lasa, et A. Hertzmann, “Trajectory optimization for full-body movements with complex contacts”, IEEE Transactions on Visualization and Computer Graphics, vol. 19, no. 8, pp. 1405–1414, Août 2013. DOI : 10.1109/TVCG.2012.325. En ligne : <http://dx.doi.org/10.1109/TVCG.2012.325>
- J. Angeles, Fundamentals of Robotic Mechanical Systems : Theory, Methods and Algorithms. Springer, 2003.
- F. Audouin et P. Stegemann, <http://robo lounge.net/>, 2013, dernière visite : 10-10-2014.
- T. Bajd, M. Mihelj, et M. Munih, Introduction to Robotics, série SpringerBriefs in Applied Sciences and Technology. Springer, 2013. En ligne : <http://books.google.ca/books?id=9ahEAAAAQBAJ>
- S. Boyd, “Sequential convex programming”, Stanford University, 2000.
- S. Boyd et L. Vandenberghe, Convex Optimization. New York, NY, USA : Cambridge University Press, 2004.
- R. W. Brockett, “Formal languages for motion description and map making”, dans Robotics, R. W. Brockett, éd., vol. 41. Providence, RI : American Mathematical Society, 1990, pp. 181–193.
- R. Burridge, A. Rizzi, et D. Koditschek, “Sequential composition of dynamically dexterous robot behaviors”, The International Journal of Robotics Research, vol. 18, no. 6, pp. 534–555, June 1999.
- S. Chiaverini, B. Siciliano, et O. Egeland, “Review of the Damped Least-Squares Inverse Kinematics with Experiments on an Industrial Robot Manipulator”, Control Systems Technology, IEEE Transactions on, vol. 2, no. 2, pp. 123–134, Jun 1994.
- J. Cooper, “Seamless mocap cycles tutorial”, <http://www.gameanim.com/2013/09/11/seamless-mocap-cycles-tutorial/>, 2013, dernière visite : 09-10-2014.

- M. Djoua, R. Plamondon, A. Della Cioppa, et A. Marcelli, “Deterministic and evolutionary extraction of delta-lognormal parameters : performance comparison”, International Journal of Pattern Recognition and Artificial Intelligence, vol. 21, no. 01, pp. 21–41, 2007.
- D. Featherstone, R. et Orin, “Robot dynamics : equations and algorithms”, Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 1, pp. 826–834, 2000. DOI : 10.1109/ROBOT.2000.844153
- F. Flacco, A. De Luca, et O. Khatib, “Prioritized multi-task motion control of redundant robots under hard joint constraints.” dans IROS. IEEE, 2012, pp. 3970–3977.
- T. Flash, “The control of hand equilibrium trajectories in multi-joint arm movements”, Biological cybernetics, vol. 57, no. 4-5, pp. 257–274, 1987.
- T. Flash et N. Hogan, “The coordination of arm movements : an experimentally confirmed mathematical model”, The journal of Neuroscience, vol. 5, no. 7, pp. 1688–1703, 1985.
- E. Frazzoli, M. A. Dahleh, et E. Feron, “Maneuver-based motion planning for nonlinear systems with symmetries”, Robotics, IEEE Transactions on, vol. 21, no. 6, pp. 1077–1091, 2005.
- K. Harada, K. Hauser, T. Bretl, et J.-C. Latombe, “Natural motion generation for humanoid robots”, dans Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE, 2006, pp. 833–839.
- R. Heck et M. Gleicher, “Parametric motion graphs”, dans Proceedings of the 2007 symposium on Interactive 3D graphics and games. ACM, 2007, pp. 129–136.
- M. Hersch et A. Billard, “Reaching with multi-referential dynamical systems.” Auton. Robots, vol. 25, no. 1-2, pp. 71–83, 2008.
- A. Hirakawa et A. Kawamura, “Trajectory generation for redundant manipulators under optimization of consumed electrical energy”, dans Industry Applications Conference, 1996. Thirty-First IAS Annual Meeting, IAS '96., Conference Record of the 1996 IEEE, vol. 3, Oct 1996, pp. 1626–1632 vol.3. DOI : 10.1109/IAS.1996.559286

- D. Hristu-Varsakelis, S. Anderson, F. Zhang, P. Sodre, et P. Krishnaprasad, “A motion description language for hybrid system programming”, Institute for Systems Research (preprint), 2003.
- L. Huo et L. Baron, “The joint-limits and singularity avoidance in robotic welding”, Industrial Robot : An International Journal, vol. 35, no. 5, pp. 456–464, 2008.
- A. J. Ijspeert, “Central pattern generators for locomotion control in animals and robots : a review”, Neural Networks, vol. 21, no. 4, pp. 642–653, 2008.
- A. J. Ijspeert, J. Nakanishi, et S. Schaal, “Movement imitation with nonlinear dynamical systems in humanoid robots”, dans Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on, vol. 2. IEEE, 2002, pp. 1398–1403.
- M. Kawato, “Trajectory formation in arm movements : minimization principles and procedures”, Advances in motor learning and control, pp. 225–259, 1996.
- B. Keiser, “Torque control of a kuka youbot arm”, Mémoire de maîtrise, Robotics and Perception Group, University of Zurich, 2013.
- O. Khatib, E. Demircan, V. De Sapio, L. Sentis, T. Besier, et S. Delp, “Robotics-based synthesis of human motion”, Journal of Physiology-Paris, vol. 103, no. 3, pp. 211–219, 2009.
- S. Kim et F. C. Park, “Fast robot motion generation using principal components : framework and algorithms”, Industrial Electronics, IEEE Transactions on, vol. 55, no. 6, pp. 2506–2516, 2008.
- D. Kirk, Optimal Control Theory : An Introduction, série Dover Books on Electrical Engineering Series. Dover Publications, 2004.
- L. Kovar, M. Gleicher, et F. Pighin, “Motion graphs”, ACM transactions on graphics (TOG), vol. 21, no. 3, pp. 473–482, 2002.
- D. Kulic, D. Lee, C. Ott, et Y. Nakamura, “Incremental learning of full body motion primitives for humanoid robots”, dans Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on. IEEE, 2008, pp. 326–332.

- D. Kulic, C. Ott, D. Lee, J. Ishikawa, et Y. Nakamura, “Incremental learning of full body motion primitives and their sequencing through human motion observation”, The International Journal of Robotics Research, p. 0278364911426178, 2011.
- J. Kwon et F. C. Park, “Natural movement generation using hidden markov models and principal components”, Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on, vol. 38, no. 5, pp. 1184–1194, 2008.
- J. Le Ny et G. Pappas, “Sequential composition of robust controller specifications”, dans Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012, pp. 5190–5195.
- A. Liegeois, “Automatic supervisory control of the configuration and behavior of multibody mechanisms”, IEEE Transactions on Systems, Man, and Cybernetics, vol. 7, no. 12, pp. 868–871, 1977.
- V. Maheu, J. Frappier, P. Archambault, et F. Routhier, “Evaluation of the jaco robotic arm : Clinico-economic study for powered wheelchair users with upper-extremity disabilities”, dans Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on, June 2011, pp. 1–5. DOI : 10.1109/ICORR.2011.5975397
- E. Orin, M.W. et Walker, “Efficient dynamic computer simulation of robotic mechanisms”, Journal of Dynamic Systems Measurement and Control-transactions of The Asme, pp. p. 205–2011, 1982.
- U. Pattacini, F. Nori, L. Natale, G. Metta, et G. Sandini, “An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots”, dans Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, 2010, pp. 1668–1674.
- D. Pescovitz, “Bot & dolly and the rise of creative robots”, <http://www.businessweek.com/articles/2014-03-20/bot-and-dolly-and-the-rise-of-creative-robots>, 2014, dernière visite : 17-10-2014.
- A. Piazzzi et A. Visioli, “Global minimum-jerk trajectory planning of robot manipulators”, Industrial Electronics, IEEE Transactions on, vol. 47, no. 1, pp. 140–149, Feb 2000. DOI : 10.1109/41.824136

- R. Plamondon et M. Djoua, “A multi-level representation paradigm for handwriting stroke generation”, Human Movement Science, vol. 25, no. 4, pp. 586–607, 2006.
- R. Plamondon et S. N. Srihari, “Online and off-line handwriting recognition : a comprehensive survey”, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 1, pp. 63–84, 2000.
- A. Robotics, Technical reference manual, RAPID Instructions, Functions and Data types, ABB, 2010, rev. J.
- , Technical Reference Manual - RAPID Overview, ABB Robotics, 2009, rev. H.
- F. Robotics, LR Mate 200iC Series & R-30iA Mate Controller, FANUC Robotics America Inc., 2009.
- C. Santiago, J. Oliveira, L. Reis, et A. Sousa, “Autonomous robot dancing synchronized to musical rhythmic stimuli”, dans Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on, June 2011, pp. 1–6.
- S. Schaal, J. Peters, J. Nakanishi, et A. Ijspeert, “Learning movement primitives”, dans Robotics Research. Springer, 2005, pp. 561–572.
- M. Seymour, “Gravity : vfx that’s anything but down to earth”, <http://www.fxguide.com/featured/gravity/>, 2013, dernière visite : 17-10-2014.
- R. Shadmehr et S. Wise, The Computational Neurobiology of Reaching and Pointing : A Foundation for Motor Learning, série A Bradford book. MIT Press, 2005.
- R. Shadmehr et F. A. Mussa-Ivaldi, “Adaptive representation of dynamics during learning of a motor task”, The Journal of Neuroscience, vol. 14, no. 5, pp. 3208–3224, 1994.
- M. Spong, S. Hutchinson, , et M. Vidyasagar, Robot Modeling and Control. John Wiley and Sons, 2006.
- J. Vince, Quaternions for Computer Graphics. Springer, 2011.

- P. Viviani et T. Flash, “Minimum-jerk, two-thirds power law, and isochrony : converging approaches to movement planning.” Journal of Experimental Psychology : Human Perception and Performance, vol. 21, no. 1, p. 32, 1995.
- Y. Wada et M. Kawato, “A via-point time optimization algorithm for complex sequential trajectory formation.” Neural networks : the official journal of the International Neural Network Society, vol. 17, no. 3, pp. 353–64, Mai 2004. DOI : 10.1016/j.neunet.2003.11.009. En ligne : <http://www.ncbi.nlm.nih.gov/pubmed/15037353>
- , “A theory for cursive handwriting based on the minimization principle”, Biological Cybernetics, vol. 73, no. 1, pp. 3–13, 1995.
- D. E. Whitney, “Resolved motion rate control of manipulators and human prostheses.” IEEE Transactions on man-machine systems, 1969.
- A. Woch et R. Plamondon, “Using the framework of the kinematic theory for the definition of a movement primitive”, MOTOR CONTROL-CHAMPAIGN-, vol. 8, pp. 547–557, 2004.
- T. Yoshikawa, Foundations of Robotics : Analysis and Control. Cambridge, MA, USA : MIT Press, 1990.
- T. Yoshioka, N. Shimada, K. Ohishi, et T. Miyazaki, “High performance final state control system based on minimum-jerk trajectory for industrial robots”, dans IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society, Nov 2011, pp. 3400–3405. DOI : 10.1109/IECON.2011.6119858

ANNEXE A

Cinématique inverse

A.1 Découplage du problème de positionnement et orientation

La CI analytique a l'avantage par rapport aux méthodes itératives d'être très rapides [Chap. 3.3](Spong et al., 2006), en plus de permettre de choisir une solution arbitraire parmi un ensemble de solutions. La CI consiste à trouver les positions articulaires permettant au robot d'atteindre une position cartésienne et une orientation donnée, c-a-d trouver $q \in \mathbb{R}^n$, n étant le nombre d'articulations, satisfaisant

$${}^0T_n(q) = \begin{pmatrix} {}^0R_n & p_{EE} \\ 0 & 1 \end{pmatrix},$$

avec $p_{EE} \in \mathbb{R}^3$ le point cartésien à atteindre et R la matrice d'orientation du repère de base au repère de l'effecteur du robot. La matrice R comporte trois composantes

$$R = \begin{pmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} \end{pmatrix},$$

\mathbf{n} étant le vecteur normal, \mathbf{o} le vecteur d'orientation et \mathbf{a} le vecteur d'approche (Fig. A.2). La plupart des manipulateurs industriels à 6 degrés de liberté (DDL), comme ceux étudiés ici, sont construits de telle sorte que les trois derniers degrés de liberté forment un poignet sphérique, c-a-d que leurs trois derniers axes s'intersectent en un point commun. On peut de cette façon résoudre séparément le problème de positionnement du problème d'orientation. Si un robot a un poignet sphérique, ses trois dernières articulations ont leur axe qui s'intersectent en un point commun, et leurs paramètres de Denavit-Hartenberg (DH) respectent la forme décrite au tableau A.1.

Les trois premières articulations du robot doivent alors atteindre le poignet

$$p = p_{EE} - d_6 \mathbf{a},$$

avec \mathbf{a} le vecteur d'approche de l'effecteur. On a dès lors la matrice de transformation découpée

$${}^0T_6^* = {}^0T_6 - \mathbf{a}d_6.$$

Tableau A.1 Forme des paramètres de DH des trois dernières articulations d'un robot avec un poignet sphérique. Les angles sont en degrés.

Joint	$a_i[m]$	$\alpha_i[rad]$	$d_i[m]$	$q_i[rad]$
$n - 2$	0	$-\frac{\pi}{2}$	0	q_4
$n - 1$	0	$\frac{\pi}{2}$	0	q_5
n	0	0	d_6	q_6

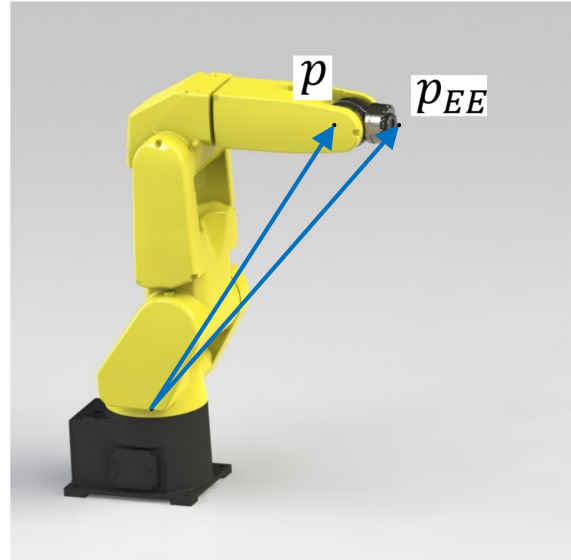
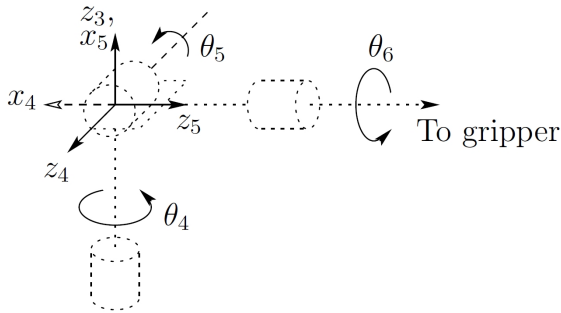


Figure A.1 Le poignet sphérique d'un robot. Les axes des trois dernières articulations, q_4 , q_5 et q_6 , s'intersectent en un point commun p qui est le poignet du robot. Cela permet de découpler le problème d'orientation et de positionnement. (Image du poignet sphérique issu de Spong et al. (2006))

La situation est illustrée en figure A.1.

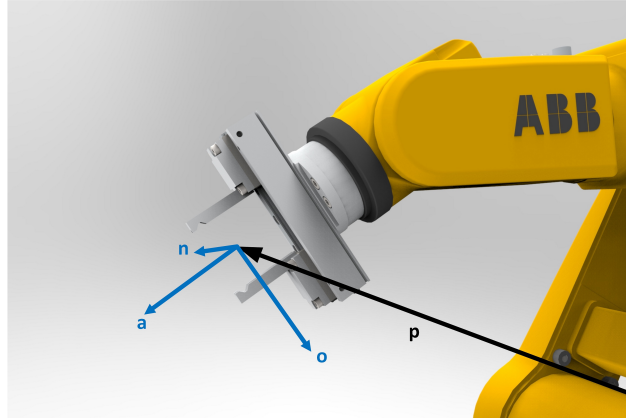


Figure A.2 Référentiel de l'effecteur. p est le vecteur position, a le vecteur d'approche, o le vecteur d'orientation et n le vecteur normal.

A.2 Cinématique inverse du FANUC LR Mate 200iC

Les paramètres de DH du FANUC LR Mate 200iC sont présentés au tableau A.2 et les repères associés à la position au repos en figure A.3.

Tableau A.2 Paramètres de Denavit-Hartenberg du FANUC LR Mate 200iC. Distances en mm, angles en degrés.

Joint	$q[rad]$	$d[m]$	$a[m]$	$\alpha[rad]$
1	q_1	0.330	0.075	$-\frac{\pi}{2}$
2	$q_2 - \frac{\pi}{2}$	0	0.300	π
3	q_3	0	0.075	$-\frac{\pi}{2}$
4	q_4	-0.320	0	$\frac{\pi}{2}$
5	q_5	0	0	$-\frac{\pi}{2}$
6	q_6	-0.080	0	π

Problème de positionnement

Le modèle inverse du robot est calculé selon les paramètres de DH montrés à la section précédente selon la technique de Spong et al. (2006). Calculons d'abord les matrices de transformation intermédiaires. Les termes de rotation sont omis ici car non nécessaires au

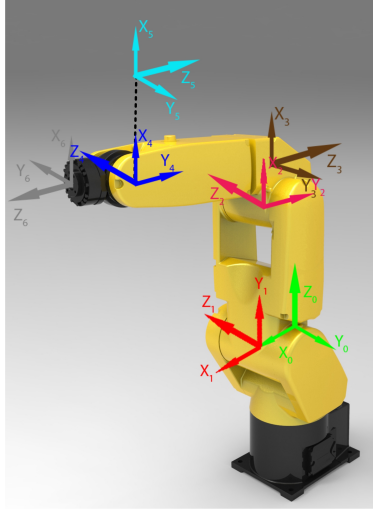


Figure A.3 FANUC LR Mate 200iC avec les repères respectifs de chaque articulation.

problème de positionnement. Les termes incluant d_6 dans les positions sont ignorés étant donné le découplage position-orientation.

$${}^0T_1^{-1} {}^0T_6^*(1 : 3, 4) = \begin{pmatrix} C_1 p_x + S_1 p_y - a_1 \\ -p_z + d_1 \\ C_1 p_y - S_1 p_x \end{pmatrix} \quad (\text{A.1})$$

$$\begin{aligned} {}^1T_6^*(1 : 3, 4) &= \begin{pmatrix} C_2 C_3 a_3 + S_2 C_3 d_4 - C_2 S_3 d_4 + S_2 S_3 a_3 + a_2 C_2 \\ C_2 S_3 d_4 + C_2 C_3 a_3 - S_2 C_3 d_4 + S_2 S_3 a_3 - a_2 S_2 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \cos(q_2 - q_3) a_3 - \sin(q_2 - q_3) d_4 + a_2 C_2 \\ \cos(q_2 - q_3) d_4 + \sin(q_2 - q_3) a_3 + a_2 S_2 \\ 0 \end{pmatrix} \end{aligned} \quad (\text{A.2})$$

$${}^0T_2^{-1} {}^0T_6^*(1 : 3, 4) = \begin{pmatrix} C_1 C_2 p_x + S_1 S_2 p_y - S_2 a_1 + C_2 p_z - C_2 d_1 - a_2 \\ -C_1 C_2 p_x - S_1 C_2 p_y + C_2 a_1 + S_2 p_z - S_2 d_1 \\ C_1 p_y - S_1 p_x \end{pmatrix} \quad (\text{A.3})$$

$${}^2T_6^*(1 : 3, 4) = \begin{pmatrix} a_3 C_3 - S_3 d_4 \\ C_3 d_4 + a_3 S_3 \\ 0 \end{pmatrix} \quad (\text{A.4})$$

avec

$${}^0T_6^* = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^0T_6 - \mathbf{ad}_6$$

qui est la matrice de transformation du repère 0 au repère 6 avec découplage du problème de positionnement et d'orientation. Pour trouver l'expression de la première articulation, on égalise les termes (3, 4) des Eqs. A.1 et A.2

$$C_1 p_y - S_1 p_x = 0. \quad (\text{A.5})$$

On peut réécrire l'équation comme

$$\frac{S_1}{C_1} = \frac{p_y}{p_x}$$

On a ainsi deux solutions possibles pour q_1 :

$$q_1 = \text{atan2}(\pm p_y, \pm p_x).$$

Pour la troisième articulation, en reprenant les termes (1, 4) et (2, 4) des Eqs. A.1 et A.2, on a

$$-a_1 + C_1 p_x + S_1 p_y = C_{2-3} a_3 - S_{2-3} d_4 + a_2 C_2 \quad (\text{A.6})$$

$$d_1 - p_z = C_{2-3} d_4 + S_{2-3} a_3 + a_2 S_2 \quad (\text{A.7})$$

avec

$$C_{2-3} = \cos(q_2 - q_3)$$

$$S_{2-3} = \sin(q_2 - q_3).$$

En effectuant le changement de variable

$$p_x = r |\cos(\phi)|$$

$$p_y = r |\sin(\phi)|$$

$$r = \pm \sqrt{p_x^2 + p_y^2}$$

$$= C_1 p_x + S_1 p_y$$

$$\phi = \text{atan2}(p_y, p_x).$$

Il est à noter qu'ici, r peut être soit positif ou négatif selon le choix posé pour q_1 . L'équation A.6 devient

$$-a_1 + r = -a_1 + r = C_{2-3}a_3 - S_{2-3}d_4 + a_2C_2. \quad (\text{A.8})$$

En mettant au carré et en additionnant les Eqs. A.8 et A.7, on obtient

$$\begin{aligned} (d_1 - p_z)^2 + r^2 + a_1^2 - 2a_1r \\ &= d_4^2 + a_3^2 + a_2^2 + 2C_{2-3}C_2a_2a_3 - 2S_{2-3}C_2a_2d_4 + C_{2-3}S_2a_2d_4 \\ &= d_4^2 + a_3^2 + a_2^2 - 2a_2d_4S_3 + 2C_3a_2a_3 \\ a_3C_3 - S_3d_4 &= \frac{r^2 + (d_1 - p_z)^2 + a_1^2 - a_2^2 - a_3^2 - d_4^2 - 2a_1r}{2a_2} \triangleq A. \end{aligned}$$

Soit le changement de variable

$$\begin{aligned} a_3 &= \rho \cos(\beta) \\ d_4 &= \rho \sin(\beta) \\ \rho &= \sqrt{a_3^2 + d_4^2} \\ \beta &= \text{atan2}(d_4, a_3). \end{aligned}$$

On trouve ainsi

$$\begin{aligned} \rho(C_3 \cos(\beta) - S_3 \sin(\beta)) &= A \\ \cos(q_3 + \beta) &= A/\rho \\ q_3 + \beta &= \text{atan2}\left(\pm \sqrt{1 - \left(\frac{A}{\rho}\right)^2}, \frac{A}{\rho}\right) \\ q_3 &= \text{atan2}\left(\pm \sqrt{1 - \left(\frac{A}{\rho}\right)^2}, \frac{A}{\rho}\right) - \text{atan2}(d_4, a_3). \end{aligned}$$

On a donc deux solutions possibles pour q_3 . Pour trouver q_2 , on utilise enfin le terme (1, 4) des Eqs. A.3 et A.4

$$\begin{aligned} -a_2 - C_2(p_z - d_1) + S_2(C_1p_x + S_1p_y - a_1) &= a_3C_3 - S_3d_4 \\ -C_2 \overbrace{(p_z - d_1)}^N + S_2 \overbrace{(C_1p_x + S_1p_y - a_1)}^H &= \overbrace{a_3C_3 - S_3d_4 + a_2}^M \end{aligned} \quad (\text{A.9})$$

et le terme (2, 4) des Eqs. A.3 et A.4

$$\begin{aligned}
-C_1 C_2 p_x - S_1 C_2 p_y + C_2 a_1 + S_2(p_z - d_1) &= C_3 d_4 + a_3 S_3 \\
-C_2(-a_1 + C_1 p_x + S_1 p_y) + S_2(p_z - d_1) &= C_3 d_4 + a_3 S_3 \\
C_2 \overbrace{(-a_1 + C_1 p_x + S_1 p_y)}^H - S_2 \overbrace{(p_z - d_1)}^N &= \overbrace{-C_3 d_4 - a_3 S_3}^L.
\end{aligned} \tag{A.10}$$

En faisant le changement de variable

$$\begin{aligned}
H &= r \cos \phi \\
N &= r \sin \phi \\
r &= \sqrt{H^2 + N^2} \\
\phi &= \text{atan2}(N, H)
\end{aligned}$$

et on obtient

$$\begin{cases} C_2 r \cos \phi - S_2 r \sin \phi &= L \\ C_2 r \sin \phi + S_2 r \cos \phi &= M \end{cases}$$

$$\begin{cases} r \cos(q_2 + \phi) &= L \\ r \sin(q_2 + \phi) &= M \end{cases}$$

ce qui donne

$$\begin{aligned}
q_2 &= \text{atan2}(M, L) - \phi \\
q_2 &= \text{atan2}(M, L) - \text{atan2}(N, H)
\end{aligned}$$

Problème d'orientation

Pour le problème d'orientation, il faut trouver les valeurs des trois dernières articulations afin de satisfaire l'orientation au bout de l'effecteur. En observant la géométrie du robot et de ses trois derniers degrés de liberté, on voit qu'on se retrouve avec la transformation

$${}^3R_6 = {}^0R_3^T {}^0R_6 = R_z(q_4)R_y(-q_5)R_z(q_6)R_x(\pi).$$

Nous sommes dans un cas d'angles d'Euler ZYZ modifié :

$$R_z(q_4)R_y(-q_5)R_z(q_6) = {}^0R_3^T {}^0R_6 R_x^T(\pi) \equiv Q.$$

La matrice de rotation associée vaut

$$\begin{aligned}
 R_z(q_4)R_y(-q_5)R_z(q_6) &= \begin{pmatrix} C_4C_5C_6 - S_4S_6 & C_6S_4 + C_4C_5S_6 & C_4S_5 \\ C_5C_6S_4 + C_4S_6 & -C_4C_6 + C_5S_4S_6 & S_4S_5 \\ C_6S_5 & S_5S_6 & -C_5 \end{pmatrix} \\
 &= \begin{pmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{pmatrix}.
 \end{aligned}$$

En calculant également

$$\begin{aligned}
 R_z(q_4)^{-1} {}^3R_6 &= \begin{pmatrix} C_4n_x + S_4n_y & C_4o_x + S_4o_y & C_4a_x + S_4a_y \\ -S_4n_x + C_4n_y & -S_4o_x + C_4o_y & -S_4a_x + C_4a_y \\ n_z & o_z & a_z \end{pmatrix} \\
 R_y(-q_5)R_z(q_6)R_x(\pi) &= \begin{pmatrix} C_5C_6 & C_5S_6 & S_5 \\ S_6 & -C_6 & 0 \\ S_5C_6 & S_5S_6 & -C_5 \end{pmatrix}
 \end{aligned}$$

De l'élément (2,3), on a

$$\begin{aligned}
 0 &= -S_4a_x + C_4a_y \\
 q_4 &= \begin{cases} \text{atan2}(a_y, a_x) \\ \text{atan2}(-a_y, -a_x) \end{cases}.
 \end{aligned}$$

Des éléments (1,3) et (3,3), on a

$$\begin{aligned}
 S_5 &= C_4a_x + S_4a_y \\
 C_5 &= -a_z \\
 q_5 &= \text{atan2}(C_4a_x + S_4a_y, -a_z)
 \end{aligned}$$

Et des éléments (2,1) et (2,2), on a

$$\begin{aligned}
 S_6 &= S_4n_x - C_4n_y \\
 C_6 &= S_4o_x - C_4o_y \\
 q_6 &= \text{atan2}(S_4n_x - C_4n_y, S_4o_x - C_4o_y)
 \end{aligned}$$

L'ensemble des solutions pour la CI du FANUC LR Mate 200iC est résumée au tableau A.3.

Tableau A.3 Solution de la CI du FANUC LR Mate 200iC. 8 solutions distinctes sont possibles.

valeur	# solutions
$q_1 = \text{atan2}(\pm p_y, \pm p_x)$	2
$q_2 = \text{atan2}(M, L) - \text{atan2}(N, H)$	1
$q_3 = \text{atan2}(\pm \sqrt{1 - \left(\frac{A}{\rho}\right)^2}, \frac{A}{\rho}) - \text{atan2}(d_4, a_3)$	2
$q_4 = \begin{cases} \text{atan2}(a_y, a_x) \\ \text{atan2}(-a_y, -a_x) \end{cases}$	2
$q_5 = \text{atan2}(C_4 a_x + S_4 a_y, -a_z)$	1
$q_6 = \text{atan2}(S_4 n_x - C_4 n_y, S_4 o_x - C_4 o_y)$	1
<hr/>	
$A = \frac{p_x^2 + p_y^2 + (d_1 - p_z)^2 + a_1^2 - a_2^2 - a_3^2 - d_4^2 - 2a\sqrt{p_x^2 + p_y^2}}{2a_2}$	
$\rho = \sqrt{a_3^2 + d_4^2}$	
$H = C_1 p_x + S_1 p_y - a_1$	
$L = -C_3 - a_3 S_3$	
$M = a_3 C_3 - S_3 d_4 + a_2$	
$N = p_z - d_1$	
$Q = {}^0R_3^T {}^0R_6 R_x(\pi)$	

A.3 Cinématique inverse du ABB IRB 120T

Les paramètres de DH du manipulateur ABB IRB 120T utilisé pour le projet (chapitre 7) sont présentés au tableau A.4 et les repères associés à la position au repos en figure A.4. La situation est quasiment identique au cas du FANUC développé ci-dessus.

Tableau A.4 Paramètres de Denavit-Hartenberg du *ABB IRB 120*. Distances en mm, angles en degrés.

Joint	$q[rad]$	$d[m]$	$a[m]$	$\alpha[rad]$
1	q_1	0.290	0	$-\frac{\pi}{2}$
2	$q_2 - \frac{\pi}{2}$	0	0.270	0
3	q_3	0	0.070	$-\frac{\pi}{2}$
4	q_4	0.374	0	$\frac{\pi}{2}$
5	q_5	0	0	$-\frac{\pi}{2}$
6	$q_6 - \pi$	0.072	0	0

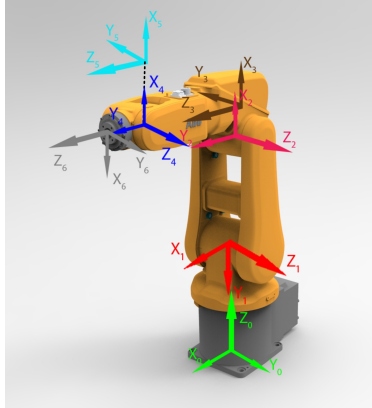


Figure A.4 ABB IRB 120T avec les repères respectifs de chaque articulation.

Problème de positionnement

Le problème de positionnement est quasiment identique à celui du FANUC. Le sens de rotation de l'axe 3 est opposé à celui du FANUC. Il en résulte que le terme L' est opposé au terme L de l'Eq. A.10. On obtient ainsi

$$L' = C_3 + a_3 S_3$$

$$q_2 = \text{atan2}(M, L') - \text{atan2}(N, H)$$

Le reste du développement est strictement identique pour le problème de positionnement.

Problème d'orientation

Pour le problème d'orientation, on observe qu'on a une rotation

$${}^3R_6 = {}^0R_3^T {}^0R_6 = R_z(q_4)R_y(-q_5)R_z(q_6)R_z(\pi).$$

Nous sommes dans un cas d'angles d'Euler ZYZ légèrement modifié :

$$R_z(q_4)R_y(-q_5)R_z(q_6) = {}^0R_3^T {}^0R_6 R_z^T(\pi) \equiv Q.$$

La matrice de rotation associée vaut

$$\begin{aligned} R_z(q_4)R_y(-q_5)R_z(q_6) &= \begin{pmatrix} -C_4C_5C_6 + S_4S_6 & C_6S_4 + C_4C_5S_6 & -C_4S_5 \\ -C_5C_6S_4 - C_4S_6 & -C_4C_6 + C_5S_4S_6 & -S_4S_5 \\ -C_6S_5 & S_5S_6 & C_5 \end{pmatrix} \\ &= \begin{pmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{pmatrix} \cdot \end{aligned}$$

En calculant également

$$\begin{aligned} R_z(q_4)^{-1} {}^3R_6 &= \begin{pmatrix} C_4n_x + S_4n_y & C_4o_x + S_4o_y & C_4a_x + S_4a_y \\ -S_4n_x + C_4n_y & -S_4o_x + C_4o_y & -S_4a_x + C_4a_y \\ n_z & o_z & a_z \end{pmatrix} \\ R_y(-q_5)R_z(q_6) &= \begin{pmatrix} C_5C_6 & -C_5S_6 & S_5 \\ S_6 & C_6 & 0 \\ -S_5C_6 & S_5S_6 & C_5 \end{pmatrix} \end{aligned}$$

De l'élément (2,3), on a

$$\begin{aligned} 0 &= -S_4a_x + C_4a_y \\ q_4 &= \begin{cases} \text{atan2}(a_y, a_x) \\ \text{atan2}(-a_y, -a_x) \end{cases} \end{aligned}$$

Des éléments (1,3) et (3,3), on a

$$S_5 = C_4 a_x + S_4 a_y$$

$$C_5 = a_z$$

$$q_5 = \text{atan2}(C_4 a_x + S_4 a_y, a_z)$$

Et des éléments (2,1) et (2,2), on a

$$S_6 = -S_4 n_x + C_4 n_y$$

$$C_6 = -S_4 o_x + C_4 o_y$$

$$q_6 = \text{atan2}(S_4 n_x - C_4 n_y, S_4 o_x - C_4 o_y)$$

L'ensemble des solutions pour la CI du ABB IRB 120T est résumée au tableau A.5.

Tableau A.5 Solution de la CI du ABB IRB 120T. 8 solutions distinctes sont possibles.

valeur	# solutions
$q_1 = \text{atan2}(\pm p_y, \pm p_x)$	2
$q_2 = \text{atan2}(M, L') - \text{atan2}(N, H)$	1
$q_3 = \text{atan2}(\pm \sqrt{1 - \left(\frac{A}{\rho}\right)^2}, \frac{A}{\rho}) - \text{atan2}(d_4, a_3)$	2
$q_4 = \begin{cases} \text{atan2}(a_y, a_x) \\ \text{atan2}(-a_y, -a_x) \end{cases}$	2
$q_5 = \text{atan2}(C_4 a_x + S_4 a_y, a_z)$	1
$q_6 = \text{atan2}(-S_4 n_x + C_4 n_y, -S_4 o_x + C_4 o_y)$	1
$A = \frac{p_x^2 + p_y^2 + (d_1 - p_z)^2 + a_1^2 - a_2^2 - a_3^2 - d_4^2 - 2a\sqrt{p_x^2 + p_y^2}}{2a_2}$	
$\rho = \sqrt{a_3^2 + d_4^2}$	
$H = C_1 p_x + S_1 p_y - a_1$	
$L' = C_3 + a_3 S_3$	
$M = a_3 C_3 - S_3 d_4 + a_2$	
$N = p_z - d_1$	
$Q = {}^0R_3^T {}^0R_6 R_z(\pi)$	

ANNEXE B

Vitesses et accélérations articulaires

B.1 Vitesses et accélérations

On a la relation

$$\dot{x} = \frac{\partial x}{\partial q} \dot{q} = J \dot{q}. \quad (\text{B.1})$$

Sous leur forme la plus simple, les vitesses articulaires s'obtiennent en remaniant l'Eq. B.1 (Bajd et al., 2013)

$$\dot{q} = J^\dagger \dot{x}, \quad (\text{B.2})$$

avec J^\dagger l'inverse généralisée (matrice pseudoinverse de Moore-Penrose) de la jacobienne du manipulateur. Les précautions quant aux problèmes de singularité sont discutées en section C. On construit la jacobienne selon Spong et al. (2006) :

Algorithm 1: Algorithme pour obtenir la matrice jacobienne J pour des joints rotoïdes

Input: n le nombre d'articulations,
 $e_z = [0 \ 0 \ 1]^T$ pour un joint sphérique,

```

 $J \leftarrow 0$ 
 $i \leftarrow n$ 
while  $i \geq 1$  do
    Calculer  ${}^{i-1}T_n$ 
     $J_i = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times {}^{i-1}p_n \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix}$ 
     $i \leftarrow i - 1$ 
    if  $i > 0$  then
         $J = \begin{bmatrix} {}^{i-1}R_i & 0 \\ 0 & {}^{i-1}R_i \end{bmatrix} J$ 
    end
return  $J$ 
end

```

En dérivant l'Eq. B.1, on obtient l'accélération

$$\begin{aligned}\ddot{x} &= \dot{J}\dot{q} + J\ddot{q} \\ \ddot{q} &= J^\dagger(\ddot{x} - \dot{J}\dot{q}).\end{aligned}$$

Afin de connaître l'accélération du manipulateur, il nous faut donc au préalable calculer la dérivée de la matrice jacobienne \dot{J} . On peut trouver $\dot{J}\dot{\theta}$ itérativement en utilisant l'algorithme itératif suivant Angeles (2003) :

Algorithm 2: Algorithme pour obtenir la matrice jacobienne \dot{J} pour des joints rotoïde

Input: n le nombre d'articulations,

```

 $\omega_1 \leftarrow \dot{\theta}e_1$ 
for  $i \leftarrow 1$  to  $n - 1$  do
  |  $\omega_{i+1} \leftarrow \dot{\omega}_{i+1}\theta_i + 1 + R_{i+1}^T\omega_i$ 
end

```

```

 $\dot{e}_1 \leftarrow 0$ 
for  $i \leftarrow 2$  to  $n$  do
  |  $\dot{e}_i \leftarrow \omega_i \times e_i$ 
end

```

```

 $\dot{r}_n \leftarrow \omega_n \times a_n$ 
for  $i \leftarrow n - 1$  to  $1$  do
  |  $\dot{r}_i \leftarrow \omega_i \times a_i + R_{i+1}\dot{r}_{i+1}$ 
end

```

```

 $\dot{u}_1 \leftarrow e_1 \times \dot{r}_1$ 
for  $i \leftarrow 2$  to  $n$  do
  |  $\dot{u}_i \leftarrow \dot{e}_i \times r_i + e_i \times \dot{r}_i$ 
end

```

```

 $U_i = \begin{bmatrix} {}^iR_{i+1} & 0 \\ 0 & {}^iR_{i+1} \end{bmatrix}$ 
 $v_n = \dot{\theta}_n \begin{bmatrix} \dot{e}_n \\ \dot{u}_n \end{bmatrix}$ 
for  $i \leftarrow n - 1$  to  $1$  do
  |  $v_i \leftarrow \dot{\theta}_i \begin{bmatrix} \dot{e}_i \\ \dot{u}_i \end{bmatrix} + U_i v_{i+1}$ 
end

```

```

 $\dot{j}\dot{\theta} = v_1$ 

```

ANNEXE C

Damped Least-Squares

Cette section détaille l'utilisation de la technique du Damped Least-Squares (DLS). Les vitesses articulaires s'obtiennent en utilisant l'Eq. B.2

$$\dot{q} = J^\dagger \dot{x}. \quad (\text{C.1})$$

L'Eq. B.2 étant vulnérable aux problèmes de singularité, il faut trouver une technique pour inverser la matrice jacobienne de façon à toujours obtenir une solution viable à proximité de singularités. On utilise l'algorithme du *Damped Least-Squares* (DLS) décrit par Chiaverini et al. (1994). Cette technique consiste à décomposer la matrice \mathbf{J} en valeurs singulières (SVD)

$$\mathbf{J} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\text{T},$$

avec $\mathbf{\Sigma}$ une matrice diagonale contenant les valeurs singulières σ_i ordonnées selon $\sigma_1 > \sigma_2 > \dots > \sigma_n$. Calculer la pseudo-inverse de \mathbf{J} revient à effectuer

$$\mathbf{J}^\dagger = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^\text{T}. \quad (\text{C.2})$$

Il s'agit donc de calculer la matrice pseudo-inverse de $\mathbf{\Sigma}$. En temps normal, la matrice étant diagonale, son expression serait réduite à

$$\mathbf{\Sigma}^\dagger = \begin{pmatrix} \frac{1}{\sigma_1} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \frac{1}{\sigma_n} \end{pmatrix}.$$

Le problème de singularité vient du fait que si on est proche d'une singularité, σ_n devient très petit et $\frac{1}{\sigma_n}$ explose. Chiaverini et al. (1994) proposent un terme d'amortissement λ permettant de transformer le terme $\frac{1}{\sigma_n}$ en un terme amorti, presque nul lorsqu'on approche d'une singularité. On évite ainsi que ce terme n'explose. Pour se faire, on introduit la pseudo-

inverse de Σ modifiée

$$\Sigma^\dagger = \begin{pmatrix} \frac{\sigma_1}{\sigma_1^2 + \lambda^2} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2}{\sigma_2^2 + \lambda^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \frac{\sigma_n}{\sigma_n^2 + \lambda^2} \end{pmatrix}$$

avec le terme d'amortissement λ :

$$\lambda^2 = \begin{cases} 0 & \text{quand } \sigma_n > \epsilon \\ (1 - (\frac{\sigma_n}{\epsilon})^2) \lambda_{\max}^2 & \text{sinon} \end{cases} \quad (\text{C.3})$$

λ_{\max} étant une valeur seuil pour λ (nous utiliserons $\lambda_{\max} = 0.04$ comme dans Chiaverini et al. (1994)) et ϵ la limite à partir de laquelle on estime être proche d'une singularité. Si on est proche d'une singularité, le terme $\frac{\sigma_n}{\sigma_n^2 + \lambda^2}$ n'explose pas et reste proche de 0. On constate également que pour $\sigma_i \gg \epsilon$, l'inversion de σ_i est peu influencée par le terme d'amortissement

$$\frac{\sigma_i}{\sigma_i^2 + \lambda^2} \approx \frac{1}{\sigma_i} \text{ si } \sigma_i \gg \epsilon.$$

ϵ est une valeur permettant de décrire si on approche une singularité. Chiaverini et Siciliano utilisent une valeur arbitraire pour ϵ . Nous allons ici remplacer ce seuil fixe par un seuil sur le nombre de condition de la matrice J , que nous appellerons ω_{\max} . Le nombre de condition ω_{cond} est connu pour être un bon indicateur de l'approche d'une singularité

$$\omega_{\text{cond}} = \frac{\sigma_1}{\sigma_n}.$$

On remplace donc le terme d'amortissement de l'Eq. C.3 par

$$\lambda^2 = \begin{cases} 0 & \text{quand } \omega_{\text{cond}} < \omega_{\max} \\ (1 - (\frac{\omega_{\max}}{\omega_{\text{cond}}})^2) \lambda_{\max}^2 & \text{sinon} \end{cases}$$

On peut passer facilement de ϵ à ω_{\max} :

$$\epsilon = \frac{\sigma_n \omega_{\text{cond}}}{\omega_{\max}} = \frac{\cancel{\sigma_n} \sigma_1}{\omega_{\max} \cancel{\sigma_n}} = \frac{\sigma_1}{\omega_{\max}},$$

la différence entre les deux critères étant que le seuil ω_{cond} utilisé pour activer le terme d'amortissement est proportionnel à σ_1 ; on active le terme d'amortissement quand une valeur propre est beaucoup plus petite que les autres, plus quand une valeur propre est simplement faible.

Plusieurs valeurs de $\omega_{condmax}$ ont été testées pour différentes trajectoires. Une trajectoire en coordonnée cartésienne est passée dans le domaine articulaire en utilisant la technique du RMR (voir section 5.2) en contraignant totalement le système. Les positions articulaires sont trouvées en utilisant la relation

$$\begin{aligned} q_{i+1} &= q_i + \Delta q \\ \Delta q &= J^\dagger \Delta x \end{aligned}$$

où J^\dagger est trouvé en utilisant la technique du DLS (Eq. C.2). Un exemple avec ses résultats est illustré en figure C.1. Le choix de la valeur de ω_{max} a été choisi expérimentalement comme

$$\omega_{max} = 1000.$$

En utilisant une valeur typique pour $\sigma_1 = 1.817$ (valeur de σ_1 à la position initiale du test en figure C.1) et en supposant σ_1 constant, cela équivaldrait à

$$\epsilon = \frac{1.817}{1000} = 0.001817.$$

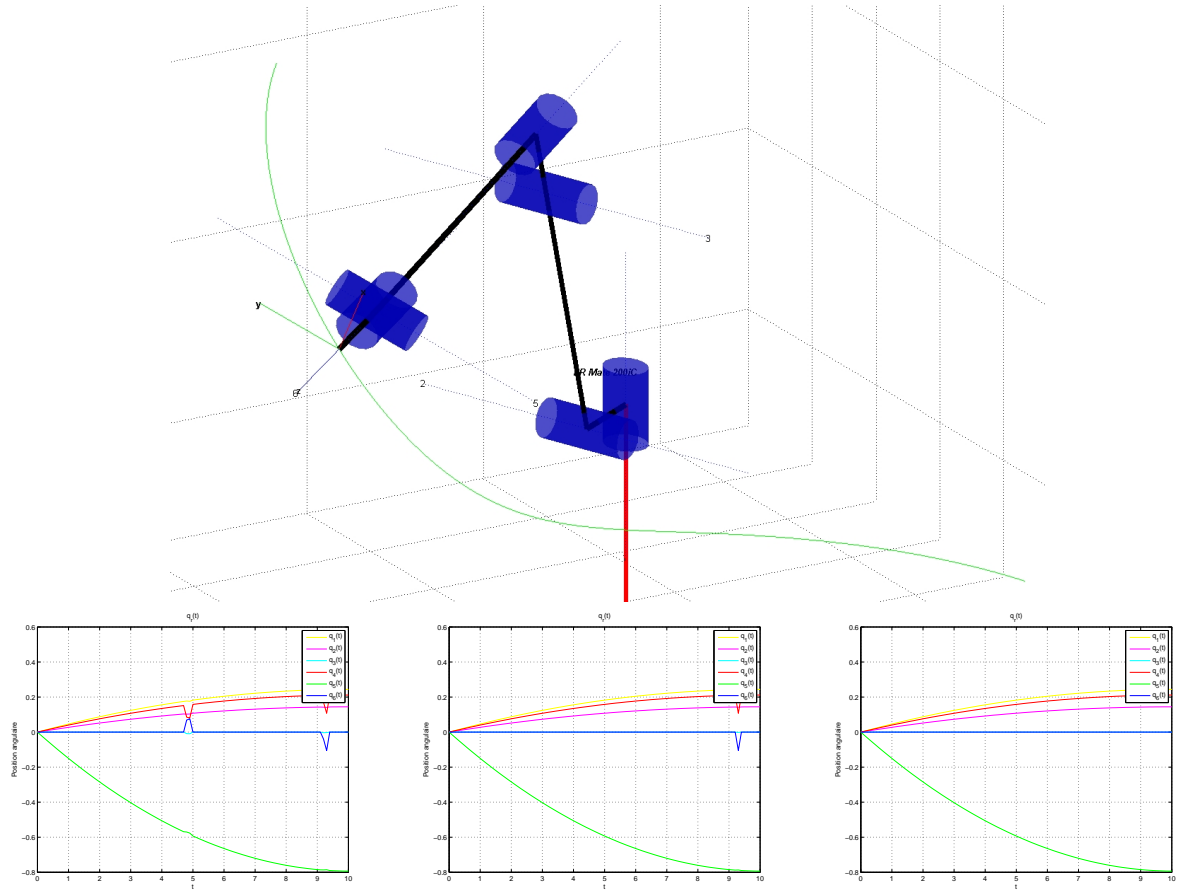


Figure C.1 Trajectoire pour l'approche de singularité avec l'algorithme de DLS (haut), et résultats de la CI (bas) pour $\omega_{\max} = 100$ (gauche), $\omega_{\max} = 500$ (milieu) et $\omega_{\max} = 1000$ (droite). $\omega_{\max} = 1000$ est la seule valeur pour laquelle la trajectoire ne dévie pas à l'approche de la singularité.

ANNEXE D

RMR - Critères d'optimisations

Cette section décrit les critères d'optimisations développés par Huo and Baron (2008) dans leur article.

D.1 RMR - Évitement des limites articulaires

Le critère pour l'évitement des limites articulaires développé par Huo and Baron (2008) s'exprime comme

$$z_{joint} = \frac{1}{2}(q - \bar{q})^T K_j^T K_j (q - \bar{q}) \rightarrow \min_q \quad (D.1)$$

avec

$$\begin{aligned} \bar{q} &= \frac{1}{2}(q_{max} + q_{min}) \\ K_j &= \text{Diag}(k_j), \end{aligned}$$

où k_j est un vecteur de coefficients pondérant le critère d'optimisation, K_j la matrice $n \times n$ avec les n coefficients de k_j sur la diagonale, q_{max} et q_{min} les positions articulaires maximale et minimale, et k_j un vecteur de coefficients pondérant le critère d'optimisation. Le gradient pour l'évitement des limites articulaires vaut

$$h_{joint} = -\nabla z_{joint} = -\frac{\partial z_{joint}}{\partial q} = K_j(\bar{q} - q). \quad (D.2)$$

D.2 RMR - Évitement des singularités

Le second critère développé par Huo and Baron (2008) est l'évitement de singularités

$$z_{sing} = \frac{\omega_{ps}^2}{2}(q - q_{TS})^T K_s^T K_s (q - q_{TS}) \rightarrow \min_q \quad (D.3)$$

avec

$$K_s = \text{Diag}(k_s)$$

où k_s est un vecteur de coefficients pondérant le critère d'optimisation. K_s est donc la matrice $n \times n$ avec les n coefficients de k_s sur sa diagonale, et l'indice

$$\omega_{ps} = \sqrt{\frac{\omega_{cond}}{\omega_{mom}}} \quad (D.4)$$

qui est un indicateur de l'approche de singularité,

$$\omega_{cond} = \frac{\sigma_1}{\sigma_m} \geq 1$$

qui est une mesure du conditionnement de la matrice J , et la mesure de manipulabilité

$$\omega_{mom} = \sqrt{\det(JJ^T)} = \sigma_1 \sigma_2 \dots \sigma_m$$

avec $\sigma_1 \sigma_2 \dots \sigma_m$ les valeurs singulières de J ordonnées du maximum au minimum. Quand ω_{ps} passe au-dessus d'un certain seuil

$$\omega_{ps} > \omega_{psmax},$$

on enregistre la position q_{TS} qui est la dernière position valide et le critère (D.3) est activé. Si on se trouve en dessous du seuil ω_{psmax} , le critère (D.3) n'est pas activé et aucune modification par rapport à la trajectoire de base n'est effectuée. Le gradient pour le critère de singularité est :

$$h_{sing} = \begin{cases} -\nabla z_{sing} = K_s \omega_{ps} (q_{TS} - q) & \text{si } \omega_{ps} > \omega_{psmax} \\ 0 & \text{sinon.} \end{cases} \quad (D.5)$$

ANNEXE E

Dynamique

Afin de contrôler un robot, il faut connaître les couples à appliquer à ses joints pour suivre une trajectoire désirée. Le chapitre précédent a permis de développer les algorithmes afin de spécifier la trajectoire à suivre en coordonnées internes q_r , \dot{q}_r et \ddot{q}_r . Dans ce chapitre, on cherche à connaître la commande :

$$u = \tau(q_r, \dot{q}_r, \ddot{q}_r, q, \dot{q})$$

qui permettra de contrôler le robot afin qu'il suive au mieux cette trajectoire. La section E.1 donne les équations de la dynamique. En section E.2 nous étudions la dynamique inverse, qui permet de trouver τ à partir de q_r , \dot{q}_r et \ddot{q}_r . La dynamique directe, expliquée en section E.3, permet quant à elle de monter un simulateur reproduisant le comportement dynamique d'un robot face à des couples articulaires donnés. Enfin, la section E.4 explique une méthode afin d'estimer les paramètres dynamiques du robot utilisé.

E.1 Équations de la dynamique

Les équations de la dynamique se trouvent en utilisant l'équation d'Euler-Lagrange (Spong et al., 2006). On en dérive l'équation de la forme générale

$$\tau = D(q)\ddot{q} + h(q, \dot{q}) + g(q) \quad (\text{E.1})$$

avec $D(q)$ la matrice d'inertie ou de masse, $h(q, \dot{q}) = C(q, \dot{q})\dot{q}$ le vecteur des forces de Coriolis et centrifuges, et $g(q)$ le vecteur des forces gravitationnelles. Deux méthodes sont possibles afin de calculer les forces articulaires de l'Eq. E.1. On peut utiliser un modèle plus complet de la dynamique prenant en compte la friction dans les joints

$$\tau = D(q)\ddot{q} + h(q, \dot{q}) + g(q) + b(\dot{q}).$$

Par manque d'information sur ces termes pour le robot réel, nous négligerons les frottements dans les joints et utiliserons l'Eq. E.1.

E.2 Dynamique inverse : calcul des forces articulaires

Le problème de dynamique inverse consiste à retrouver τ à partir de q , \dot{q} et \ddot{q} . Deux techniques permettent de réaliser cela. La première technique consiste à utiliser les équations de Lagrange pour dériver les forces articulaires. La seconde technique est l'algorithme de Newton-Euler récursif (NER). Comme expliqué dans Yoshikawa (1990, chap. 3.5), la formulation de Lagrange a une complexité en $\mathcal{O}(n^4)$ tandis que l'algorithme de NER a une complexité en $\mathcal{O}(n)$ et est donc plus efficace en terme de temps de calcul. Nous utiliserons donc la formulation de NER.

L'algorithme de Newton-Euler récursif pour des joints sphériques comme expliqué dans Yoshikawa (1990) est illustré à l'algorithme 3. Il est à noter que si des forces ou couples non nuls sont appliqués sur l'effecteur, les valeurs de f_{n+1} et n_{n+1} ne sont dès lors plus nulles mais correspondent à la force et au couple exercés sur l'effecteur, respectivement.

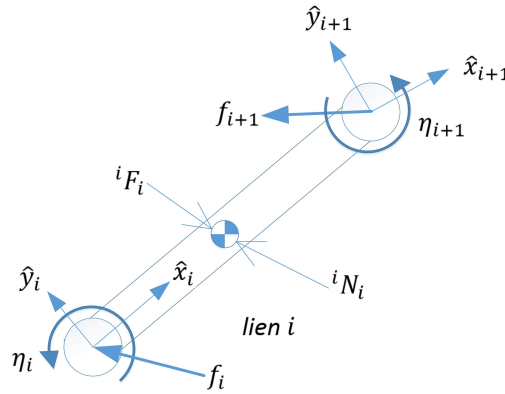


Figure E.1 Schéma des différents termes apparaissant dans l'algorithme de Newton-Euler récursif.

Il est possible (non fait ici) d'améliorer la rapidité de l'algorithme de NER (dans une optique d'optimisation pour une application temps réel) grâce à différentes techniques listées dans Featherstone (2000), et en particulier dans l'article de M.W Walker et D.E Orin Orin (1982).

E.3 Dynamique directe : calcul des accélérations articulaires

Afin de pouvoir simuler le comportement du robot, il est nécessaire de savoir déterminer les mouvements du robot résultant des couples appliqués à ses articulations. Il faut pour cela

Algorithm 3: Algorithme de Newton-Euler récursif pour n joints sphérique

Input: $g = 9.81$ la force de gravité exercée sur le manipulateur,
 $e_z = [0 \ 0 \ 1]^T$ pour un joint sphérique,
 ω_i la vitesse angulaire que subit le joint i ,
 ${}^{i-1}\mathbf{R}_i$, la matrice de rotation permettant de passer du repère $i - 1$ au repère i ,
 ${}^i p_{i+1}$ le vecteur, exprimé dans le repère i reliant le repère i au repère $i + 1$,
 s_i la position du centre de masse du joint i ,
 I_i l'inertie du joint i ,
 F_i la force exercée exclusivement sur le centre de gravité du joint i ,
 N_i le couple exercé exclusivement sur le centre de gravité du joint i ,
 f_i la force totale exercée sur le joint i ,
 n_i le couple total exercé sur le joint i ,
 τ_i le couple à appliquer au joint i afin de générer la trajectoire désirée.

```

 $\omega_0 \leftarrow 0$ 
 $\dot{\omega}_0 \leftarrow 0$ 
 $\ddot{p}_0 \leftarrow [0 \ 0 \ -g]^T$ 
for  $i \leftarrow 1$  to  $n$  do
     $\omega_i \leftarrow {}^{i-1}\mathbf{R}_i^T \omega_i + \mathbf{e}_z \dot{q}_i$ 
     $\ddot{\omega}_i \leftarrow {}^{i-1}\mathbf{R}_i^T \dot{\omega}_{i-1} + \mathbf{e}_z + ({}^{i-1}\mathbf{R}_i^T \omega_{i-1}) \times \mathbf{e}_z \dot{q}_i$ 
     $\ddot{p}_i \leftarrow {}^{i-1}\mathbf{R}_i^T [\ddot{p}_{i-1} + \dot{\omega}_i \times {}^{i-1}\mathbf{p}_i + \omega_{i-1} \times (\omega_{i-1} \times {}^{i-1}\mathbf{p}_i)]$ 
end
 $f_{n+1} \leftarrow 0$ 
 $n_{n+1} \leftarrow 0$ 
for  $i \leftarrow n$  to  $1$  do
     $\ddot{s}_i \leftarrow \ddot{p}_i + \dot{\omega}_i \times s_i + \omega_i \times (\omega_i \times s_i)$ 
     $F_i \leftarrow m_i \ddot{s}_i$ 
     $N_i \leftarrow I_i \omega_i + \omega_i \times (I_i \omega_i)$ 
     $f_i \leftarrow {}^i\mathbf{R}_{i-1} \mathbf{f}_{i+1} + \mathbf{F}_i$ 
     $n_i \leftarrow {}^i\mathbf{R}_{i+1} \mathbf{n}_{i+1} + \mathbf{N}_i + \mathbf{s}_i \times \mathbf{F}_i + {}^i\mathbf{p}_{i+1} \times ({}^i\mathbf{R}_{i+1} \mathbf{f}_{i+1})$ 
     $\tau_i \leftarrow e_z^T n_i$ 
end
return  $\underline{\tau}$ 

```

réaliser un modèle de dynamique directe du robot : il s'agit de trouver \ddot{q} à partir de τ . Comme expliqué dans Yoshikawa (1990, chap. 3.5), on peut réécrire l'Eq. E.1 comme

$$\begin{aligned} D(q)\ddot{q} &= \tau - \tau_N \\ \tau_N &= h(q, \dot{q}) = C(q, \dot{q})\dot{q}. \end{aligned}$$

où τ_N est trouvé en utilisant NER en imposant $\ddot{q} = 0$. La matrice d'inertie $D(q)$ a la forme

$$D(q) = [D_1(q) \ D_2(q) \ \dots \ D_j(q)]$$

où $D_j(q)$ est trouvé en imposant $\dot{q} = 0$, une gravité nulle ($g(q) = 0$) et en imposant une accélération unitaire à l'axe j , i.e.

$$\ddot{q} = e_j,$$

où e_j est le vecteur unitaire pour lequel toutes les valeurs sont nulles sauf la j^{me} valeur

$$e_j(j) = 1.$$

La matrice d'inertie $D(q)$ étant définie positive, elle est toujours inversible et on peut écrire

$$\ddot{q} = D^{-1}(q)[M^{-1}(q)[\tau - h(q, \dot{q}) - g(q)]$$

ce qui nous permet de trouver l'accélération articulaire. On intègre encore afin de retrouver \dot{q} et q successivement.

$$\begin{aligned} \ddot{q} &= M^{-1}(q)[\tau - h(q, \dot{q}) - g(q)] \\ \dot{q} &= \int_{t_0}^{t_f} \ddot{q} dt \\ q &= \int_{t_0}^{t_f} \dot{q} dt, \end{aligned}$$

Ce qui se traduit dans une simulation numérique par un développement en série de Taylor d'ordre 2 en Δt

$$\begin{aligned} \ddot{q}(t) &= M^{-1}(q(t))[\tau - h(q(t), \dot{q}(t)) - g(q(t))] \\ \dot{q}(t + \Delta t) &= \dot{q}(t) + \ddot{q}(t)\Delta t \\ q(t + \Delta t) &= q(t) + \dot{q}(t)\Delta t + \ddot{q}(t)(\Delta t)^2/2. \end{aligned}$$

Nous utiliserons ici le simulateur multicorps *Simmechanics* afin de simuler le comportement du robot.

E.4 Détermination des paramètres dynamiques

Comme vu à la section E.2, nous avons besoin de connaître différentes données sur le robot comme l'inertie de ses articulations et leur centre de masse respective afin de pouvoir simuler le robot efficacement. Un modèle 3D du robot *FANUC LR Mate 200iC* est repris du site GRABCAD. Le modèle est repris dans le logiciel de CAD *Catia* et on donne à chaque articulation du robot un poids proportionnel à son volume (on suppose une répartition uniforme du poids dans le volume de chaque articulation), en s'assurant que le poids du robot (27kg (Robotics, 2009b)) est respecté. *Catia* permet de récupérer la position des centres de masse ainsi que la matrice d'inertie. Les paramètres dynamiques obtenus sont illustrés au tableau E.1.

Tableau E.1 Paramètres dynamiques du FANUC LR Mate 200iC obtenu à partir d'un modèle 3D du robot avec le logiciel de CAD *Catia*.

Joint	Masse [kg]	Centre de gravité [m]	Matrice d'inertie [kg × m ²]
1	4.54	$\begin{pmatrix} 22.476 \times 10^{-3} \\ -0.798 \times 10^{-3} \\ -85.215 \times 10^{-3} \end{pmatrix}$	$\begin{pmatrix} 4.512 \times 10^{-4} & 4.512 \times 10^{-4} & -7 \times 10^{-3} \\ 4.512 \times 10^{-4} & 2.7 \times 10^{-2} & 8.033 \times 10^{-4} \\ -7 \times 10^{-3} & 8.033 \times 10^{-4} & 3.9 \times 10^{-2} \end{pmatrix}$
2	6	$\begin{pmatrix} -1.41 \times 10^{-4} \\ 11.56 \times 10^{-3} \\ 110 \times 10^{-3} \end{pmatrix}$	$\begin{pmatrix} 8.5 \times 10^{-2} & 2.86 \times 10^{-5} & -1.738 \times 10^{-4} \\ 2.86 \times 10^{-5} & 6.8 \times 10^{-2} & -4 \times 10^{-3} \\ -1.738 \times 10^{-4} & -4 \times 10^{-3} & 3 \times 10^{-2} \end{pmatrix}$
3	2.711	$\begin{pmatrix} 2.46 \times 10^{-4} \\ 6.38 \times 10^{-4} \\ 37.874 \times 10^{-3} \end{pmatrix}$	$\begin{pmatrix} 9 \times 10^{-3} & -4.136 \times 10^{-6} & -2 \times 10^{-3} \\ -4.136 \times 10^{-6} & 1 \times 10^{-2} & -3.788 \times 10^{-5} \\ -2 \times 10^{-3} & -3.788 \times 10^{-5} & 8 \times 10^{-3} \end{pmatrix}$
4	2.156	$\begin{pmatrix} -128.714 \times 10^{-3} \\ 4.306 \times 10^{-3} \\ 2.33 \times 10^{-4} \end{pmatrix}$	$\begin{pmatrix} 4 \times 10^{-3} & -1.339 \times 10^{-4} & 7.177 \times 10^{-6} \\ -1.339 \times 10^{-4} & 1.1 \times 10^{-2} & 5.79 \times 10^{-6} \\ 7.177 \times 10^{-6} & 5.79 \times 10^{-6} & 5.79 \times 10^{-6} \end{pmatrix}$
5	0.447	$\begin{pmatrix} 15.307 \times 10^{-3} \\ 1.624 \times 10^{-3} \\ -2.251 \times 10^{-5} \end{pmatrix}$	$\begin{pmatrix} 3.084 \times 10^{-4} & 1.264 \times 10^{-5} & 0 \\ 1.264 \times 10^{-5} & 6.027 \times 10^{-4} & 0 \\ 0 & 0 & 5.491 \times 10^{-4} \end{pmatrix}$
6	0.013	$\begin{pmatrix} -1.73 \times 10^{-4} \\ 0 \\ -6.386 \times 10^{-3} \end{pmatrix}$	$\begin{pmatrix} 1.56 \times 10^{-6} & 0 & 0 \\ 0 & 1.523 \times 10^{-6} & 0 \\ 0 & 0 & 2.78 \times 10^{-6} \end{pmatrix}$

Nous utilisons ces informations ainsi que le modèle 3D directement dans *Simmechanics* afin de simuler la dynamique du robot. Cette technique permet d'obtenir une approximation des paramètres dynamiques du robot. Le schéma *Simmechanics* ainsi que son aperçu 3D est illustré en figure E.2.

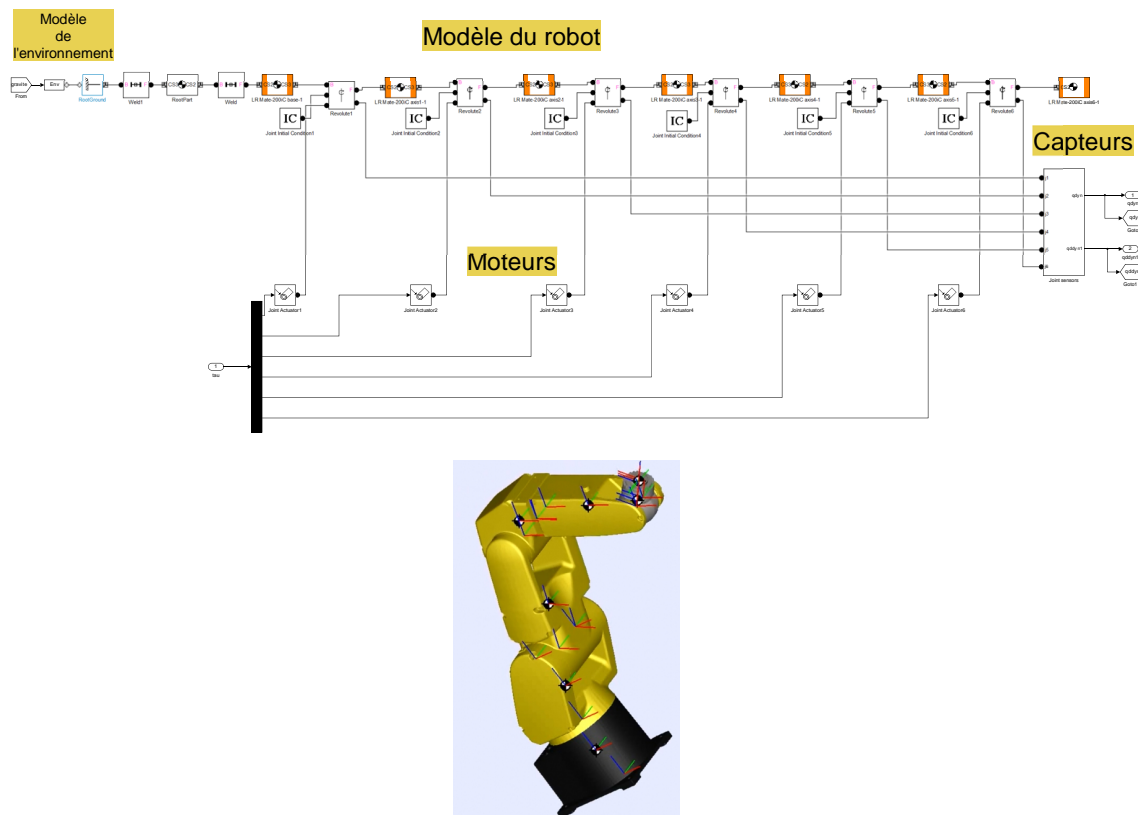


Figure E.2 Schéma Simmechanics (haut) et aperçu 3D (bas) du FANUC LR Mate 200iC.

ANNEXE F

Quaternions

F.1 Notions de base

Afin de pouvoir interpoler les orientations et générer des transitions de façon cohérente, il est nécessaire d'exprimer les orientations en terme de quaternions. La théorie sur les quaternions est résumée dans Vince (2011). Nous appellerons dans ce document un quaternion m au lieu de l'appellation usuelle q car q désigne dans ce document les positions articulaires. Un quaternion a la forme

$$m = [s, x\mathbf{i} + y\mathbf{j} + z\mathbf{k}].$$

On peut réexprimer m comme

$$m = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{v} \right]$$

où on a l'axe \mathbf{v} et l'angle de rotation θ . Nous ne ferons mention ici que de quaternions unitaires

$$|m| = 1.$$

Le conjugué d'un quaternion a la forme

$$m^{-1} = \left[\cos\left(\frac{\theta}{2}\right), -\sin\left(\frac{\theta}{2}\right)\mathbf{v} \right]$$

et le produit entre quaternions vaut

$$q_0 q_1 = [s_0 s_1 - \mathbf{v}_0 \cdot \mathbf{v}_1, s_0 \mathbf{v}_1 + s_1 \mathbf{v}_0 + \mathbf{v}_0 \times \mathbf{v}_1].$$

F.2 Interpolation de quaternions

Afin de passer d'un quaternion q_0 à q_1 , on effectue une interpolation *Slerp* (spherical linear interpolation). Le Slerp permet d'interpoler linéairement sur une sphère deux quaternions pour obtenir un troisième quaternion intermédiaire. Contrairement à l'interpolation de vecteurs, l'interpolation de quaternions est difficilement visualisable. On interpole deux quaternions en

effectuant

$$m(t) = \frac{\sin((1-t)\Omega)}{\sin(\Omega)} q_0 + \frac{\sin(t\Omega)}{\sin(\Omega)} q_1 \quad (\text{F.1})$$

avec

$$\Omega = \arccos(q_0 \cdot q_1)$$

et $t \in [0, 1]$. L'avantage du Slerp est qu'il interpole linéairement l'angle entre les deux quaternions de norme unitaire, ce qui donne une vitesse angulaire constante entre ceux-ci. Le slerp est illustré en figure F.1.

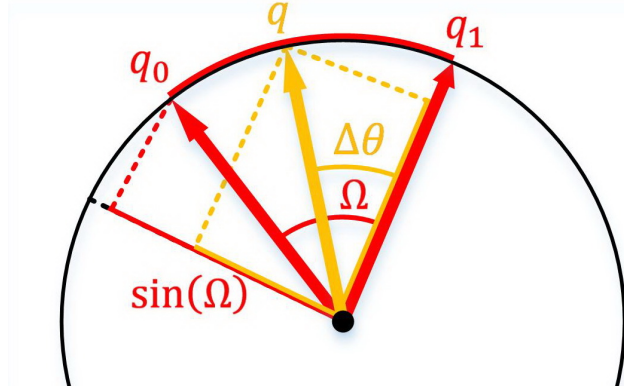


Figure F.1 Principe du Slerp. Le Slerp permet d'avoir une vitesse angulaire constante entre les deux quaternions interpolés.

Un quaternion peut être vu comme une exponentielle complexe

$$\begin{aligned} m &= \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right) \mathbf{v} \right] \\ &= \exp\left(\frac{\theta}{2} \mathbf{v}\right). \end{aligned} \quad (\text{F.2})$$

Une expression équivalente à l'Eq. F.1 est dès lors

$$m(t) = q_0 (q_0^{-1} q_1)^t.$$

On peut dériver cette expression pour trouver les dérivées du Slerp et les relier à la vitesse

angulaire. On dérive l'Eq. F.2

$$\begin{aligned}
 \dot{m}(t) &= \log(q_0^{-1}q_1) (q_0(q_0^{-1}q_1)^t) \\
 &= \frac{1}{2} [(\theta_1 \mathbf{v}_1) - (\theta_0 \mathbf{v}_0)] m(t) \\
 &= \frac{1}{2} (\Delta\theta \mathbf{v}) m(t) \\
 \Delta\theta \mathbf{v} &= (\theta_1 \mathbf{v}_1) - (\theta_0 \mathbf{v}_0)
 \end{aligned}$$

Le vecteur $\Delta\theta \mathbf{v}$ n'est rien d'autre que la vitesse angulaire ω

$$\omega = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}.$$

On peut dès lors écrire

$$\begin{aligned}
 \dot{m}(t) &= \frac{1}{2} W(t) m(t) \\
 W(t) &= \log(q_0^{-1}q_1) \\
 &= 2\dot{m}(t)m(t)
 \end{aligned} \tag{F.3}$$

avec $W(t)$ le quaternion pur de la vitesse angulaire $\omega(t)$

$$W(t) = [0, \omega_x(t), \omega_y(t), \omega_z(t)]$$

on observe que $W(t)$ et la vitesse angulaire ω est constante et dépend de q_0 et q_1 . L'Eq. F.3 est valable pour $t \in [0, 1]$ (durée $t_f = 1$). Si $t_f \neq 1$, la vitesse angulaire sera mise à l'échelle d'un facteur t_f :

$$\begin{aligned}
 W(t/t_f) &= \frac{1}{t_f} \log(q_0^{-1}q_1) \\
 &= \frac{1}{t_f} 2\dot{m}(t)m(t).
 \end{aligned}$$

F.3 Passage des angles d'Euler aux quaternions

Afin de passer des angles d'Euler ZYX aux quaternions, on effectue les trois rotations des angles d'Euler en utilisant leur quaternion respectif :

$$\begin{aligned} q_x &= \left[\cos\left(\frac{\alpha_x}{2}\right), \sin\left(\frac{\alpha_x}{2}\right)\mathbf{i} \right] \\ q_y &= \left[\cos\left(\frac{\alpha_y}{2}\right), \sin\left(\frac{\alpha_y}{2}\right)\mathbf{j} \right] \\ q_z &= \left[\cos\left(\frac{\alpha_z}{2}\right), \sin\left(\frac{\alpha_z}{2}\right)\mathbf{k} \right]. \end{aligned}$$

On utilise ensuite le quaternion résultant de la rotation simultanée des trois angles d'Euler ZYX

$$m = q_z q_y q_x.$$

La démarche est reprise dans Vince (2011, chap. 7.11) pour des angles d'Euler différents. En exprimant le quaternion sous la forme

$$m = [s, x\mathbf{i} + y\mathbf{j} + z\mathbf{k}],$$

on obtient l'expression du quaternion m en fonction des angles d'Euler

$$s = \cos \frac{\alpha_x}{2} \cos \frac{\alpha_y}{2} \cos \frac{\alpha_z}{2} + \sin \frac{\alpha_x}{2} \sin \frac{\alpha_y}{2} \sin \frac{\alpha_z}{2} \quad (\text{F.4})$$

$$x = \sin \frac{\alpha_x}{2} \cos \frac{\alpha_y}{2} \cos \frac{\alpha_z}{2} - \cos \frac{\alpha_x}{2} \sin \frac{\alpha_y}{2} \sin \frac{\alpha_z}{2} \quad (\text{F.5})$$

$$y = \cos \frac{\alpha_x}{2} \sin \frac{\alpha_y}{2} \cos \frac{\alpha_z}{2} + \sin \frac{\alpha_x}{2} \cos \frac{\alpha_y}{2} \sin \frac{\alpha_z}{2} \quad (\text{F.6})$$

$$z = \cos \frac{\alpha_x}{2} \cos \frac{\alpha_y}{2} \sin \frac{\alpha_z}{2} - \sin \frac{\alpha_x}{2} \sin \frac{\alpha_y}{2} \cos \frac{\alpha_z}{2}. \quad (\text{F.7})$$

En dérivant cette expression, on arrive à exprimer la vitesse angulaire de chacun des termes du quaternion en fonction des angles d'Euler et de leurs vitesses

$$\begin{aligned}
\dot{s} &= -\left(\frac{1}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \dot{\alpha}_x + \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_x \\
&\quad - \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \dot{\alpha}_y + \frac{1}{2} \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_y \\
&\quad + \frac{1}{2} \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \dot{\alpha}_z - \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_z \\
\dot{x} &= \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_x + \frac{1}{2} \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_x \\
&\quad - \frac{1}{2} \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \dot{\alpha}_y - \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_y \\
&\quad - \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \dot{\alpha}_z - \frac{1}{2} \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_z \\
\dot{y} &= -\left(\frac{1}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \dot{\alpha}_x + \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_x \\
&\quad + \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_y - \frac{1}{2} \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_y \\
&\quad + \frac{1}{2} \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \dot{\alpha}_z - \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_z \\
\dot{z} &= -\left(\frac{1}{2}\right) \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \dot{\alpha}_x - \frac{1}{2} \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_x \\
&\quad - \frac{1}{2} \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \dot{\alpha}_y - \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_y \\
&\quad + \frac{1}{2} \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_z + \frac{1}{2} \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_z
\end{aligned}$$

et enfin, les accélérations angulaires de chacun des termes du quaternion

$$\begin{aligned}
\ddot{s} = & \frac{1}{4} \left(-\left(\cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_y}{2}\right)\cos\left(\frac{\alpha_z}{2}\right) + \sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\right)\dot{\alpha}_x^2 \right. \\
& - \left(\cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_y}{2}\right)\cos\left(\frac{\alpha_z}{2}\right) + \sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\right)\dot{\alpha}_y^2 \\
& + 2\left(\cos\left(\frac{\alpha_y}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_x}{2}\right) + \cos\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\right)\dot{\alpha}_y\dot{\alpha}_z \\
& - \cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_y}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)\dot{\alpha}_z^2 - \sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\dot{\alpha}_z^2 \\
& + 2\dot{\alpha}_x\left(\left(\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right) + \cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\right)\dot{\alpha}_y \right. \\
& + \left.\left(\cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_y}{2}\right) + \cos\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\right)\dot{\alpha}_z \right) \\
& - 2\cos\left(\frac{\alpha_y}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_x}{2}\right)(\ddot{\alpha}_x) + 2\cos\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)(\ddot{\alpha}_x) \\
& - 2\cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)(\ddot{\alpha}_y) + 2\cos\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)(\ddot{\alpha}_y) \\
& + 2\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)(\ddot{\alpha}_z) - 2\cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)(\ddot{\alpha}_z) \Big) \\
\ddot{x} = & \frac{1}{4} \left(\left(-\cos\left(\frac{\alpha_y}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_x}{2}\right) + \cos\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\right)\dot{\alpha}_x^2 \right. \\
& + \left(-\cos\left(\frac{\alpha_y}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_x}{2}\right) + \cos\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\right)\dot{\alpha}_y^2 \\
& + 2\left(-\cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_y}{2}\right)\cos\left(\frac{\alpha_z}{2}\right) + \sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\right)\dot{\alpha}_y\dot{\alpha}_z \\
& - \cos\left(\frac{\alpha_y}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_x}{2}\right)\dot{\alpha}_z^2 + \cos\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\dot{\alpha}_z^2 \\
& - 2\dot{\alpha}_x\left(\left(\cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_y}{2}\right) - \cos\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\right)\dot{\alpha}_y \right. \\
& + \left.\left(-\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right) + \cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)\right)\dot{\alpha}_z \right) \\
& + 2\cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_y}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)(\ddot{\alpha}_x) + 2\sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)(\ddot{\alpha}_x) \\
& - 2\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)(\ddot{\alpha}_y) - 2\cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)(\ddot{\alpha}_y) \\
& - 2\cos\left(\frac{\alpha_x}{2}\right)\cos\left(\frac{\alpha_z}{2}\right)\sin\left(\frac{\alpha_y}{2}\right)(\ddot{\alpha}_z) - 2\cos\left(\frac{\alpha_y}{2}\right)\sin\left(\frac{\alpha_x}{2}\right)\sin\left(\frac{\alpha_z}{2}\right)(\ddot{\alpha}_z) \Big)
\end{aligned}$$

$$\begin{aligned}
\ddot{y} = & \frac{1}{4} \left(-\left(\cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) + \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_z}{2}\right)\right) \dot{\alpha}_x^2 \right. \\
& - \left(\cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) + \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_z}{2}\right)\right) \dot{\alpha}_y^2 \\
& - 2\left(\cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) + \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right)\right) \dot{\alpha}_y \dot{\alpha}_z \\
& - \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \dot{\alpha}_z^2 - \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_z^2 \\
& - 2\dot{\alpha}_x \left(\left(\cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) + \cos\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right)\right) \dot{\alpha}_y \right. \\
& - \left(\cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) + \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right)\right) \dot{\alpha}_z \Big) \\
& - 2 \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) (\ddot{\alpha}_x) + 2 \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) (\ddot{\alpha}_x) \\
& + 2 \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) (\ddot{\alpha}_y) - 2 \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) (\ddot{\alpha}_y) \\
& + 2 \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) (\ddot{\alpha}_z) - 2 \cos\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) (\ddot{\alpha}_z) \Big) \\
\ddot{z} = & \frac{1}{4} \left(\left(\cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) - \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right)\right) \dot{\alpha}_x^2 \right. \\
& + \left(\cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) - \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right)\right) \dot{\alpha}_y^2 \\
& + 2\left(-\cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) + \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_z}{2}\right)\right) \dot{\alpha}_y \dot{\alpha}_z \\
& + \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \dot{\alpha}_z^2 - \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) \dot{\alpha}_z^2 \\
& - 2\dot{\alpha}_x \left(\left(\cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) - \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right)\right) \dot{\alpha}_y \right. \\
& + \left(\cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) - \cos\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right)\right) \dot{\alpha}_z \Big) \\
& - 2 \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) (\ddot{\alpha}_x) - 2 \cos\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) (\ddot{\alpha}_x) \\
& - 2 \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) \sin\left(\frac{\alpha_x}{2}\right) (\ddot{\alpha}_y) - 2 \cos\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) (\ddot{\alpha}_y) \\
& + 2 \cos\left(\frac{\alpha_x}{2}\right) \cos\left(\frac{\alpha_y}{2}\right) \cos\left(\frac{\alpha_z}{2}\right) (\ddot{\alpha}_z) + 2 \sin\left(\frac{\alpha_x}{2}\right) \sin\left(\frac{\alpha_y}{2}\right) \sin\left(\frac{\alpha_z}{2}\right) (\ddot{\alpha}_z) \Big)
\end{aligned}$$

F.4 Passage des quaternions aux angles d'Euler

Malgré leurs désavantages, les angles d'Euler sont un moyen visuel efficace pour exprimer une rotation dans l'espace. Afin d'exprimer la matrice jacobienne J de façon habituelle, il est également important de pouvoir repasser des quaternions aux angles d'Euler. En observant

les Eqs. F.4, F.5, F.6 et F.7, on peut voir que

$$\begin{aligned}
2(xy + sz) &= \cos \alpha_x \sin \alpha_z \\
ss + xx - yy - zz &= \cos \alpha_x \cos \alpha_z \\
-2(xz - sy) &= \sin \alpha_x \\
2(yz + sx) &= \cos \alpha_x \sin \alpha_y \\
ss - xx - yy + zz &= \cos \alpha_x \cos \alpha_y.
\end{aligned}$$

On peut donc exprimer les angles d'Euler ZYX à partir de ces expressions :

$$\alpha_x = \begin{cases} \arcsin(-2(xz - sy)) \\ \pi - \arcsin(-2(xz - sy)) \end{cases} \quad (\text{F.8})$$

$$\alpha_y = \text{atan2}\left(\frac{2(yz + sx)}{\cos \alpha_x}, \frac{ss - xx - yy + zz}{\cos \alpha_x}\right) \quad (\text{F.9})$$

$$\alpha_z = \text{atan2}\left(\frac{2(xy + sz)}{\cos \alpha_x}, \frac{ss + xx - yy - zz}{\cos \alpha_x}\right), \quad (\text{F.10})$$

à la condition que $\alpha_x \neq \pm \frac{\pi}{2}$.

On a donc toujours deux solutions d'angles d'Euler possibles pour un même quaternion. À noter qu'on se trouve dans un cas de Gimbal lock quand $\alpha_x = \pm \frac{\pi}{2}$, alors que ce problème n'existe pas quand on utilise uniquement la représentation des quaternions.

Grâce à un logiciel de calcul symbolique, on peut exprimer les vitesses et accélérations angulaires des angles d'Euler en fonction des vitesses du quaternion équivalent. Là encore, deux choix de vitesses existent selon le choix effectué pour les angles d'Euler en Eqs. F.8, F.9 et F.10. Leur valeurs sont données ici à titre indicatif. Pour $\alpha_x = \arcsin(-2(xz - sy))$, α_y et α_z qui en découlent,

$$\begin{aligned}
\dot{\alpha}_x &= (2(-x^3\dot{s} + s^3\dot{x} + x(-y^2\dot{s} + z^2\dot{s} + 2yz\dot{x}) - (y^2 + z^2)(-z\dot{y} + y\dot{z}) - x^2(z\dot{y} + y\dot{z}) \\
&\quad + s^2(-x\dot{s} + z\dot{y} + y\dot{z}) + s(x^2\dot{x} - y^2\dot{x} + z^2\dot{x} + y(-2z\dot{s} + 2x\dot{y}) - 2xz\dot{z}))) \\
&\quad / (s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2) \\
\dot{\alpha}_y &= (2(y\dot{s} - x\dot{z} + s\dot{y} - x\dot{z}))/\sqrt{(1 - 4(sy - xz)^2)} \\
\dot{\alpha}_z &= (2(x^2(z\dot{s} - y\dot{x}) - (y^2 + z^2)(z\dot{s} + y\dot{x}) + x^3\dot{y} + s^2(-z\dot{s} + y\dot{x} + x\dot{y}) + s^3\dot{z} \\
&\quad + x(y^2\dot{y} - z^2\dot{y} + 2yz\dot{z}) + s(-2x(y\dot{s} + z\dot{x}) + 2yz\dot{y} + x^2\dot{z} - y^2\dot{z} + z^2\dot{z}))) \\
&\quad / (s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2),
\end{aligned}$$

et pour les accélérations,

$$\begin{aligned}
\ddot{\alpha}_x = & (2(-4(s^3\dot{s} + 2xyz\dot{s} + s(x^2 - y^2 + z^2)\dot{s} + x^3\dot{x} + 2syxz\dot{x} + x(y^2 - z^2)\dot{x} + 2sxyz\dot{y} \\
& + 2sxyz\dot{z} + x^2(y\dot{y} - z\dot{z}) + s^2(x\dot{x} - y\dot{y} + z\dot{z}) + (y^2 + z^2)(y\dot{y} + z\dot{z}))(-x^3\dot{s} \\
& + s^3\dot{x} + x(-y^2\dot{s} + z^2\dot{s} + 2yz\dot{x}) - (y^2 + z^2)(-z\dot{y} + y\dot{z}) - x^2(z\dot{y} + y\dot{z}) \\
& + s^2(-x\dot{s} + z\dot{y} + y\dot{z}) + s(x^2\dot{x} - y^2\dot{x} + z^2\dot{x} + y(-2z\dot{s} + 2x\dot{y}) - 2xz\dot{z})) \\
& + (s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2) \\
& (s^3\ddot{x} - y^2(2\dot{s}\dot{x} + 2\dot{y}\dot{z} + x\ddot{z} - z\ddot{y}) - (x^2 - z^2)(2\dot{s}\dot{x} + 2\dot{y}\dot{z} + x\ddot{z} + z\ddot{y}) \\
& - y^3\ddot{z} + s^2(2\dot{s}\dot{x} + 2\dot{y}\dot{z} - x\ddot{z} + z\ddot{y} + y\ddot{z}) - y(2z(\dot{s}^2 - \dot{x}^2 - \dot{y}^2 + \dot{z}^2 - x\ddot{x}) + x^2\ddot{z} + z^2\ddot{z}) \\
& + s(-2yz\ddot{z} + x^2\ddot{x} - y^2\ddot{x} + z^2\ddot{x} - 2x(\dot{s}^2 - \dot{x}^2 - \dot{y}^2 + \dot{z}^2 - y\ddot{y} + z\ddot{z})))) \\
& / (s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2) \\
\ddot{\alpha}_y = & (2(4(sy - xz)(y\dot{s} - z\dot{x} + s\dot{y} - x\dot{z})^2 + (1 - 4(sy - xz)^2)(2\dot{s}\dot{y} - 2\dot{x}\dot{z} + y\ddot{z} - z\ddot{x} + s\ddot{y} - x\ddot{z}))) \\
& / (1 - 4(sy - xz)^2)^{(3/2)} \\
\ddot{\alpha}_z = & (2(-4(s^3\dot{s} + 2xyz\dot{s} + s(x^2 - y^2 + z^2)\dot{s} + x^3\dot{x} + 2syxz\dot{x} + x(y^2 - z^2)\dot{x} + 2sxyz\dot{y} + 2sxyz\dot{z} \\
& + x^2(y\dot{y} - z\dot{z}) + s^2(x\dot{x} - y\dot{y} + z\dot{z}) + (y^2 + z^2)(y\dot{y} + z\dot{z}))(x^2(z\dot{s} - y\dot{x}) - (y^2 + z^2)(z\dot{s} + y\dot{x}) \\
& + x^3\dot{y} + s^2(-z\dot{s} + y\dot{x} + x\dot{y}) + s^3\dot{z} + x(y^2\dot{y} - z^2\dot{y} + 2yz\dot{z}) \\
& + s(-2x(y\dot{s} + z\dot{x}) + 2yz\dot{y} + x^2\dot{z} - y^2\dot{z} + z^2\dot{z})) + (s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) \\
& + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2)(s^2(2\dot{x}\dot{y} + 2\dot{s}\dot{z} - z\ddot{z} + y\ddot{x}) \\
& - (y^2 + z^2)(2\dot{x}\dot{y} + 2\dot{s}\dot{z} + z\ddot{z} + y\ddot{x}) + x^3\ddot{y} + s^3\ddot{z} + x^2(2\dot{x}\dot{y} + 2\dot{s}\dot{z} + z\ddot{z} - y\ddot{x} + s\ddot{z}) \\
& + s(-2z(\dot{s}^2 + \dot{x}^2 - \dot{y}^2 - \dot{z}^2 - y\ddot{y}) - y^2\ddot{z} + z^2\ddot{z}) + x(-2sz\ddot{x} + s^2\ddot{y} + y^2\ddot{y} - z^2\ddot{y} \\
& - 2y(\dot{s}^2 + \dot{x}^2 - \dot{y}^2 - \dot{z}^2 + s\ddot{z} - z\ddot{z})))) \\
& / (s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2)^2.
\end{aligned}$$

Pour $\alpha_x = \pi - \arcsin(-2(xz - sy))$, α_y et α_z qui en découlent, on a les vitesses

$$\begin{aligned}\dot{\alpha}_x &= ((2(-x^3\dot{s} + s^3\dot{x} + x(-y^2\dot{s} + z^2\dot{s} + 2yz\dot{x}) - (y^2 + z^2)(-z\dot{y} + y\dot{z}) \\ &\quad - x^2(z\dot{y} + y\dot{z}) + s^2(-x\dot{s} + z\dot{y} + y\dot{z}) + s(x^2\dot{x} - y^2\dot{x} + z^2\dot{x} \\ &\quad + y(-2z\dot{s} + 2x\dot{y}) - 2xz\dot{z}))/ (s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) \\ &\quad + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2) \\ \dot{\alpha}_y &= ((2(y\dot{s} - z\dot{x} + s\dot{y} - x\dot{z}))/\sqrt{1 - 4(sy - xz)^2} \\ \dot{\alpha}_z &= ((2(x^2(z\dot{s} - y\dot{x}) - (y^2 + z^2)(z\dot{s} + y\dot{x}) + x^3\dot{y} + s^2(-z\dot{s} + y\dot{x} + x\dot{y}) \\ &\quad + s^3\dot{z} + x(y^2\dot{y} - z^2\dot{y} + 2yz\dot{z}) + s(-2x(y\dot{s} + z\dot{x}) + 2yz\dot{y} + x^2\dot{z} \\ &\quad - y^2\dot{z} + z^2\dot{z}))/ (s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2),\end{aligned}$$

et pour les accélérations,

$$\begin{aligned}
\ddot{a}_x = & (2(-4(s^3\dot{s} + 2xyz\dot{s} + s(x^2 - y^2 + z^2)\dot{s} + x^3\dot{x} + 2syxz\dot{x} + x(y^2 - z^2)\dot{x} \\
& + 2sxyz\dot{y} + 2sxy\dot{z} + x^2(y\dot{y} - z\dot{z}) + s^2(x\dot{x} - y\dot{y} + z\dot{z}) + (y^2 + z^2)(y\dot{y} \\
& + z\dot{z}))(-x^3\dot{s} + s^3\dot{x} + x(-y^2\dot{s} + z^2\dot{s} + 2yz\dot{x}) - (y^2 + z^2)(-z\dot{y} \\
& + y\dot{z}) - x^2(z\dot{y} + y\dot{z}) + s^2(-x\dot{s} + z\dot{y} + y\dot{z}) + s(x^2\dot{x} - y^2\dot{x} \\
& + z^2\dot{x} + y(-2z\dot{s} + 2x\dot{y}) - 2xz\dot{z})) + (s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) \\
& + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2)(s^3\ddot{x} - y^2(2\dot{s}\dot{x} + 2\dot{y}\dot{z} + x\ddot{s} - z\ddot{y}) \\
& - (x^2 - z^2)(2\dot{s}\dot{x} + 2\dot{y}\dot{z} + x\ddot{s} + z\ddot{y}) - y^3\ddot{z} + s^2(2\dot{s}\dot{x} + 2\dot{y}\dot{z} \\
& - x\ddot{s} + z\ddot{y} + y\ddot{z}) - y(2z(\dot{s}^2 - \dot{x}^2 - \dot{y}^2 + \dot{z}^2 - x\ddot{x}) + x^2\ddot{z} \\
& + z^2\ddot{z}) + s(-2yz\ddot{s} + x^2\ddot{x} - y^2\ddot{x} + z^2\ddot{x} - 2x(\dot{s}^2 - \dot{x}^2 - \dot{y}^2 + \dot{z}^2 \\
& - y\ddot{y} + z\ddot{z}))))/(s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2)^2 \\
\ddot{a}_y = & (2(4(sy - xz)(y\dot{s} - z\dot{x} + s\dot{y} - x\dot{z})^2 + (1 - 4(sy - xz)^2)(2\dot{s}\dot{y} \\
& - 2\dot{x}\dot{z} + y\ddot{s} - z\ddot{x} + s\ddot{y} - x\ddot{z}))/((1 - 4(sy - xz)^2)^{(3/2)} \\
\ddot{a}_z = & (2(-4(s^3\dot{s} + 2xyz\dot{s} + s(x^2 - y^2 + z^2)\dot{s} + x^3\dot{x} + 2syxz\dot{x} + x(y^2 - z^2)\dot{x} \\
& + 2sxyz\dot{y} + 2sxy\dot{z} + x^2(y\dot{y} - z\dot{z}) + s^2(x\dot{x} - y\dot{y} + z\dot{z}) + (y^2 + z^2)(y\dot{y} \\
& + z\dot{z}))(x^2(z\dot{s} - y\dot{x}) - (y^2 + z^2)(z\dot{s} + y\dot{x}) + x^3\dot{y} + s^2(-z\dot{s} + y\dot{x} \\
& + x\dot{y}) + s^3\dot{z} + x(y^2\dot{y} - z^2\dot{y} + 2yz\dot{z}) + s(-2x(y\dot{s} + z\dot{x}) + 2yz\dot{y} \\
& + x^2\dot{z} - y^2\dot{z} + z^2\dot{z})) + (s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) + 2s^2(x^2 - y^2 + z^2) \\
& + (y^2 + z^2)^2)(s^2(2\dot{x}\dot{y} + 2\dot{s}\dot{z} - z\ddot{s} + y\ddot{x}) - (y^2 + z^2)(2\dot{x}\dot{y} + 2\dot{s}\dot{z} \\
& + z\ddot{s} + y\ddot{x}) + x^3\ddot{y} + s^3\ddot{z} + x^2(2\dot{x}\dot{y} + 2\dot{s}\dot{z} + z\ddot{s} - y\ddot{x} + s\ddot{z}) \\
& + s(-2z(\dot{s}^2 + \dot{x}^2 - \dot{y}^2 - \dot{z}^2 - y\ddot{y}) - y^2\ddot{z} + z^2\ddot{z}) + x(-2sz\ddot{x} \\
& + s^2\ddot{y} + y^2\ddot{y} - z^2\ddot{y} - 2y(\dot{s}^2 + \dot{x}^2 - \dot{y}^2 - \dot{z}^2 + s\ddot{s} - z\ddot{z})))) \\
& /(s^4 + x^4 + 8sxyz + 2x^2(y^2 - z^2) + 2s^2(x^2 - y^2 + z^2) + (y^2 + z^2)^2)^2.
\end{aligned}$$